On the Number of Distributed Measurement Points for Network Tomography

Joseph D. Horton
Faculty of Computer Science
University of New Brunswick
Fredericton, NB E3B 5A3, Canada
jdh@unb.ca

Alejandro López-Ortiz
School of Computer Science
University of Waterloo
Waterloo, Ont. N2L 3G1, Canada
alopez-o@uwaterloo.ca

ABSTRACT

Internet topology information is only made available in aggregate form by standard routing protocols. Connectivity information and latency characteristics must therefore be inferred using indirect techniques. In this paper we consider measurements using a distributed set of measurement points or beacons. We show that computing the minimum number of required beacons on a network under a BGP-like routing policy is NP-hard and at best $\Omega(\log n)$-approximable. In the worst case at least $(n-1)/3$ and at most $(n+1)/3$ beacons are required for a network with n nodes. We then introduce some observations that allow us to propose a relatively small candidate set of beacons for the current Internet topology. The set proposed has properties with relevant applications for all-paths routing on the public Internet and performance based routing.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**] Network monitoring

General Terms

Theory, Measurements

Keywords

Network measurements, Internet tomography, topology discovery, NP-hard, approximation algorithms, resilient overlay networks.

1. INTRODUCTION

Efficient routing and caching require accurate connectivity information of the Internet. However, by their very nature, Internet protocols make this task difficult. Routing decisions are made locally and most often shared across organizations only in aggregate form. Furthermore connectivity changes dynamically due to node or link failures and router misconfiguration. At any given time between 1.5% and 3.4% of connections suffer a visible pathology [29]. Empirically, it has been observed that a few key failures often have significant impact on routing decisions.

Routing decisions and content distribution networks (web caches) require proper connectivity and latency information so as to direct traffic in an optimal fashion. The family of internet protocols collect and distribute only a limited amount of information on the topology, connectivity and state of the network. Hence the information of interest, be it latency, topology, or connectivity has to be inferred from experimental measurements. Gathering connectivity information through indirect measurements is known as Internet tomography [15, 37, 14]. In this work we consider the problem of determining the topology of the Internet under the assumption that a distributed set of measurement points (sometimes called beacons) running special software is deployed at key sites across the entire Internet. This has emerged as one of the strategies of choice for measuring the state of the Internet (e.g. [33, 7, 3, 36, 18]).

1.1 Related Work

There are two distinctive types of measurements that can be obtained through the use of tomography: (1) obtain an accurate map of the slowly evolving link topology of the network, and (2) detect short-lived, transient effects. For the first objective, we can use long lived processes, spawning perhaps several days, while for the second we need a fast and accurate method of detecting changes, with as light a load as possible on the network.

Currently there are several efforts in progress to obtain topology and performance measurements on the Internet [33, 26, 7, 37, 3, 36, 18], several of which use some form of measurement points to extract information from the network. In practice these measurement points are often placed in universities and other organizations that are willing to host the software or hardware required. The location of these measurement devices or beacons is determined according to various heuristics [1, 4, 10, 13, 7]

Extensive research has taken place over the last few years on deploying measurement points and studying their characteristics. For example the National Internet Measurement Infrastructure (NIMI) [1, 2, 3, 31, 18] is a concerted effort to deploy general purpose beacons, termed "NIMI probes" with particular focus in scalability and flexibility. Some other measurement efforts of note are, in no particular order, MINC [12], the Internet Weather Report [33], Cheswick et al. visualization project [13], Claffy et al. efforts on internet tomography [15, 14], SPAND [35], Malan and Jahandian's Windmill [27], as well as a set of performance measurements that relied implicitly on a distributed measurement architecture (e.g. [16, 21, 22, 29, 30, 20]).

While substantial efforts have been directed at the deployment and use of distributed measurement systems, there has been somewhat less research focused on the systematic study of the properties required for such measurement sets. Jamin et al. propose theoretical methods as well as ad hoc heuristics for computing the location of a set of measurement points whose aim is to compute the distance maps on the network [25]. Recently, Barford et al. provided the first systematic experimental study to validate the empirical observation that a relatively small number of measurement points is generally sufficient to obtain an accurate map of the network [8]. Lastly, Bu et al. consider the problem of the effectiveness of tomography on networks of general topology [11]. While their focus is on the ability to infer performance data from a set of multicast trees, their systematic study of the abilities of tomography in arbitrary networks is germane to our work, as we also consider the placement of measurement points both in the current internet topology and in general networks of arbitrary topology.

In this paper we study the optimal and systematic placement of these beacons and the properties of a beacon set mapping the network both under theoretical and empirical analysis.

First we take a theoretical tack and prove in Section 4 that computing the minimum number of required beacons is NP-hard under a BGP-like routing policy on a general network. Using a reduction to minimum set cover we prove that at best this problem is $\Omega(\log n)$-approximable. We show that in terms of the number of nodes on the network, in the worst case at least $(n-1)/3$ and at most $(n+1)/3$ beacons are required for a network with n nodes. Then in Sections 5 and 6 we use an empirical approach building upon measurement data published elsewhere in the literature to show that placing beacons on a few thousand specially selected nodes suffices to map the current Internet. This seems to be within the range of feasibility for a large connectivity-sensitive organization such as a content distribution network (CDNs). Moreover, it follows from the analysis that by placing special tunnelling nodes on higher arity nodes it is possible to route over all possible paths on the Internet, thus allowing to overlay an arbitrary routing protocol on top of the public Internet forming a Resilient Overlay Network (RON) [5].

2. THE MODEL

Consider a computer network, such as the Internet, in which every node can transmit a data message to any other with proper acknowledgement if successful. That is, in its proper state the network is connected.

We model the network as an undirected graph. Hosts correspond to nodes and links to edges. Every node in the network can apply local routing policy decisions. However, those routing policies are such that the network is connected in its proper state, e.g. the root of a tree cannot refuse to carry transit traffic from one branch to another regardless of local routing policy.

The edges are labelled with non-negative weights indicating some metric such as latency or AS-hop distance. The path taken by a message can be determined at the source. In particular, in the case of the Internet this can be obtained separately through a `traceroute` call.

BGP supports a variety of mechanisms to establish routing policy. Two of the most common are AS-hop path length heuristic and administrator defined preferences. We consider two routing models: (a) arbitrary routing, which reflects a network in which all policy decisions are made based on arbitrary local preferences and (b) link distance or AS-hop length minimization. In the earlier case, in each node of higher arity, the administrator declares a single preferred route whenever more than one choice is available, and in the latter case a distance minimization (BGP- or OSPF-like) routing policy is assumed. Interestingly, in practice, while the network is not fully AS-hop metric routed, AS-path prepending is often used for implementing ad-hoc routing policy, rather than LOCAL-PREF [17, 9]. AS-path prepending preserves most AS-hop metric properties, so our results apply in that scenario as well.

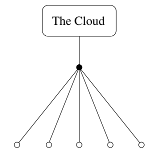

Figure 1: Degree k, arity 1.

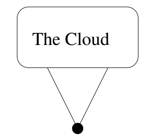

Figure 2: Degree 2, arity 2.

In this paper we consider a BGP-like routing policy in which weights attached along a given path are non-decreasing as distance increases. A node may set a local preference policy by which one path is preferred over another regardless of weight or may choose not to broadcast available connectivity to a node if an alternate path is known to be available. We assume that when forwarding a message, a node does not route a message back to the path from the sender to itself, unless it has already tried all alternative routes and determined that there was no transit path through any of its other neighbours to the destination[1]. In the latter case the message is sent back towards the node from whence it came.

We consider networks in which the routing policy at each node is applied consistently. That is, the routing policy is not changing in an adversarial fashion. The network behaves as expected with the exception of links that are down, which are presumed to be in that state for a non-instantaneous time duration. More formally, if node v is to send a message to node u, v will always first try to use the same link (edge) on the network. This implies that for any pair of nodes (v, u), whenever all edges are usable, the path to be followed by a message sent from v for u is uniquely determined.

3. BASIC CONCEPTS

On the Internet, a collection of nodes under a single routing policy and running under a single technical administration is called an Autonomous System (AS) [23]. Informally, an AS is said to be multihomed if it is directly connected to more than one national service provider (NSP). We consider a generalization of this concept to nodes that are not necessarily border routers as well as multiple provider points, even to the same ISP. An ordered pair of nodes u and v are said to exhibit *arity* m if there are precisely m edges incident with u which are connected to v other than through u. The arity of a node u is the maximum over all other nodes v of the arity exhibited by the pair u and v.

Notice that the arity of a node is never greater than the degree of the node. Furthermore as the network is connected, every node can be reached through at least one path, therefore every node has arity of at least one. Figure 1 shows a node of degree k that yet has a unique choice for each message routing operation and hence is of arity 1. In contrast Figure 2 shows a node of degree 2 and arity 2.

A node with arity $m \geq 2$ is said to be of *higher arity*.

The notion of arity introduces an interesting classification on routers. Consider a network with redundant paths to a destination from a given router. That is, a higher arity node. Higher arity nodes necessitate a *routing policy* to determine in which of several valid directions to forward a message. That is, in a network where all nodes are of arity 1 there is no need for a routing policy (although distribution of paths is still a required function). Observe that multihomed networks contain nodes of higher arity.

A node u is said to *offer transit* if, for any node v which exhibits

[1] In practice the sender can achieve this by acting as if the route through itself had been withdrawn.

higher arity from node u, then whenever u can send a message to v via an edge (u, q) then q can send a message to v via u. This reflects the standard internet usage in which multihomed nodes are said to offer transit only if they provide external access to the multiplicity of paths. The routing on a network is said to be *monotonic* if a subpath of any route path is also a route path. In other words, let $(u_0, u_1, \ldots, u_n)$ be the path taken by a message from a node u_0 to a node u_n. Then the path from u_0 to u_i, with $1 \leq i < n$ is given by the first $i + 1$ nodes in the original path from u_0 to u_n. In practice, BGP routing is not necessarily monotonic.

4. PLACING BEACONS: THEORY

We consider a network in which a given link might become unavailable but otherwise routing policy remains consistent. This is certainly the case for short spans in the Internet where even if the link topology is known at a given point in time, beacons are still needed to learn about changes in connectivity due to misconfigurations and failures. In this paper all that a beacon can do is send a message to other nodes in the network and see what route the message takes. However, it is assumed that it has control over its local routing policy and its own network interfaces so it can send a message through any incident edge (link) regardless of the default routing policy. In our abstraction, the hardware layer provides bidirectional connectivity over a physical link.

DEFINITION 1. *The* Beacon Placement Problem *is to determine the minimum number (and/or position) of beacons on a network of known topology and routing policy so that for every edge in the network there exists a sequence of messages originating from nodes of the beacon set that can determine if a given edge is up or down.*

DEFINITION 2. *Given any edge, if there exists at least one beacon which can generate a message that must transit that edge on its path to the destination and otherwise the transmission fails, then such a placement of beacons is called a* beacon set.

CLAIM 1. *A necessary and sufficient condition for a collection of beacon nodes to determine if any single arbitrary edge in a monotonic network is down is the capability to transit each edge in the network with a message originating in some beacon.*

PROOF. Assume that the beacon set transits all edges in the network, then if a single edge goes down, the beacon set can send a message that transits that edge under normal conditions and compare the new traceroute information with the old traceroute information. Then a sequence of probes is issued to determine if each of the other edges in the path are alive, which because of the monotonicity condition will necessarily succeed.

The beacon set must transit all edges, as otherwise if the precise edge that is not traversed goes down then the beacon set has no way of testing the edge and detecting this fact. □

Given two nodes n and m, define the *route path* $RP(n, m)$, to be the path followed by messages sent from n to m. Let $RPST(n)$ be the union of the $RP(n, m)$ over all possible nodes m.

If the network topology is known at start time, say through the use of an earlier round of tomography, the $RPST(n)$ for a given computer n can be found by doing a breadth first search from n, sorting the edges at each node in increasing distance. Alternatively, if no global map of the network is to be had, it can be computed by probing the IP address space from each beacon node via each of it's neighbouring nodes on the network.

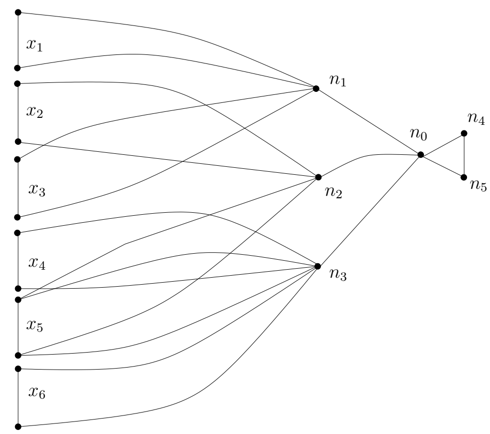

Figure 3: Set covering reduction to beacon placement.

In the AS-hop length minimization case, the $RPST(n)$ is the shortest path spanning tree rooted at n, and can be calculated using Dijstra's shortest path algorithm.

Either way, once the $RPST(n)$ tree has been computed we can use dynamic probing to detect any single dynamic edge failure on the network. More formally,

CLAIM 2. *A node n can determine by polling whether any given single edge of* $\mathrm{RPST}(n)$ *or* $\mathrm{RPST}(m)$ *are down, where m is a neighbour of n.*

PROOF. Assume that there is at most one edge in the network that is down. The node n can determine if an edge (x, y) of $RPST(n)$ is down, assuming that no other edge is down. The node n sends messages to all nodes in the $RPST(n)$ in a breadth-first search and compares the path used against the $RPST(n)$. A difference in the paths traversed indicates a failed edge. If (x, y) is not down, and no other edge on $RP(n, y)$ is down, then the acknowledgement from y tells n that (x, y) was used in sending the message and cannot be down. If (x, y) is not down but some other edge on $RP(n, y)$ is down, then we can find this out by progressively probing the nodes along the original path until a difference is observed. This edge is down, and by the assumption that at most one edge is down, (x, y) is down only if this is the edge where these paths first differ. The edges in $RPST(m)$ can be probed using an analogous procedure by n sending the message requests to m first. □

To reduce the expense of establishing a beacon set, a worthy goal is to minimize the number of beacon nodes required. We show that under a shortest path model this problem is NP-hard.

THEOREM 1. *The Beacon Placement Problem is NP-hard.*

PROOF. First observe that given a set of k nodes which form a candidate set and a description of the network and its routing policy one can readily verify that the set of k nodes is a beacon set, which shows that the Beacon Placement problem is within the NP class.

To prove hardness we construct a transformation from Minimum Set Cover which is known to be NP-complete [19]. In this problem an instance is a collection of sets $S_1, S_2, \ldots, S_m \subset U = \{x_1, \ldots, x_n\}$ and the objective is to obtain a subcollection T of k sets or less, such that jointly they contain U, i.e. $\cup_{S_i \in T} S_i = U$.

The transformation is as follows: for each $x_i \in U$, define an edge $e_i = \{v_i, u_i\}$ of weight 1 . For each S_j, define a node n_j.

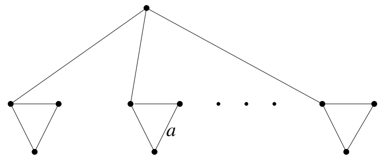

Figure 4: A network requiring $(n-1)/3$ beacons (link distance).

Connect the edges $\{n_j, v_i\}$ and $\{n_j, u_i\}$ if and only if $x_i \in S_j$. Let these edges have weight 2. Add three more nodes n_0, n_{m+1}, and n_{m+2}. Join n_0 to all the n_j for $j = 1, \ldots, m+2$. The nodes n_0, n_{m+1} and n_{m+2} form a triangle with edges weighted 1. This is illustrated with an example in Figure 3, in which $S_1 = \{x_1, x_3\}$, $S_2 = \{x_2, x_5\}$ and $S_3 = \{x_4, x_5, x_6\}$.

Routing is under a BGP-like policy with AS-hop distance metric. It follows then that edges in the n_0, n_{m+1} and n_{m+2} triangle can only be tested if one of those nodes is a beacon. Since n_0 is the only node connected to the rest of the network and the triangle is otherwise symmetric then the optimal placement of a beacon in that triangle is n_0. With a beacon thus placed we have that all edges (n_0, n_i) for $1 \leq i \leq m+2$ are testable. Edges (n_i, u_j) and (n_i, v_j) are also testable from n_0 by means of sending a message to u_j or v_j through the link to n_i.

The only edges that remain to be tested are then of the form $\{v_i, u_i\}$ corresponding to a set element x_i. These edges are part of a triangle composed by v_i, u_i and a node n_j. Therefore they can only be tested by placing a beacon at any of these three points. Lastly, if in each triangle we move the beacon from a node u_i or v_i to n_j the testability of the network remains the same, and moreover, the collection of beacons on the nodes n_j, with $1 \leq j \leq m$ form a covering set on the minimum covering set problem. Then $n_0 \cup \{n_j \mid S_j \in T\}$ is a beacon set if and only if T is a set cover of U. □

COROLLARY 1. *The Beacon Placement Problem has no approximation algorithm with a ratio better than $\Omega(\log n)$.*

PROOF. Note that the transformation maps each set to a distinct beacon. Moreover, as shown by Raz and Safra [32] there is no approximation algorithm for Minimum Set Cover with a performance ratio better than $c \log n$ for a constant $c > 0$. □

CLAIM 3. *Any connected network of n computers requires at most $(n+1)/3$ computers and may require up to $(n-1)/3$ beacon nodes under a link distance minimization routing policy.*

PROOF. To prove the upper bound, consider any arbitrary network. Choose any node in the network as the root of a depth-first-search tree. Every edge is now either a tree edge, joining a node to its child, or a back edge, leading from a node to an ancestor in the tree. Label each node in the DFS-tree by its distance to the root in the tree (number of ancestors), and reduced modulo 3, with the root labelled 0, its children are labelled 1, etc. To make the proof work, label the root 2 as well as 0.

Every edge in the tree joins a node v labelled i to a node labelled $(i+1)$ mod 3. The parent of v is labelled $(i-1)$ mod 3, unless v is the root. Therefore every edge in the tree is within one edge of a node with any given label. This is true even when v is the root, because the root is labelled 2 as well as 0.

Now consider the other edges in the network, the back edges of the DFS-tree. The ancestor is labelled i, its parent is labelled $i-1$ (if it is the root, it itself is labelled $i-1$) and its child is labelled $i+1$. Thus for any edge in the network, and for any label $i = 0$, 1 or 2, either one endpoint of the edge is labelled i or one of the endpoints is adjacent to a node labelled i.

From this last observation one can show that the nodes labelled i, for $i = 0$, 1 or 2, form a beacon set. Consider an edge (u, v). If either u or v is labelled i, then that node can test it directly. Otherwise, there is a node n labelled i that is adjacent to u or v, say u. Then n sends a message to u for v. By the link distance minimization routing policy, u must send it directly to v. If it does not, then the edge must be down. Since the number of labels is $n+1$, some label occurs no more than $(n+1)/3$ times. This completes the proof for the upper bound.

For the lower bound, consider the network shown in Figure 4. Edge a can only be tested via a message sent from one of the vertices of the triangle. Analogously, we can apply the same argument to all other triangles in the network and hence every triangle must contain a beacon node. There are $(n-1)/3$ such triangles from which the lower bound follows. □

In the case of an arbitrary routing policy, a network with n nodes may require as many as $n-2$ beacon nodes. To verify this consider a network where the links form a complete graph. The routing policy is such that the default route used by all nodes is a cycle containing all the nodes. Now, by way of contradiction assume there is a beacon set with strictly less than $n-2$ beacons. This means there are at least three nodes which are not beacons. These three nodes taken together are connected by a triangle, as the graph is complete. At most two of the triangle edges are in the default-routing cycle. Then if this third edge is down, and no other edge is down, there is no node outside its two end points who could send data on it. Hence in the worst case as many as $n-2$ nodes are required to test an arbitrary network.

Indeed this suggests, perhaps not at all surprisingly, that care must be taken when designing a network so that is readily testable from a few selected measurement points. This is also yet another argument against deploying a complete network: not only is it expensive to build and maintain, a complete network is also expensive to test and accurately diagnose.

5. PLACING BEACONS: PRACTICE

In the previous section we showed that, in general, beacons are not a very cost effective method for discovering the topology of an unknown arbitrary network. This demarcates the limits of the effectiveness of beacon sets in general. On the other hand, the internet is far from being an arbitrary network. Hence in this section we blend theoretical properties from the previous section with knowledge of the special properties of the internet to obtain an effective procedure to select a beacon set.

CLAIM 4. *Placing a beacon on every node of higher arity (if there are any) forms a beacon set in a network in which every node of higher arity offers transit so long as the network remains connected.*

PROOF. Consider an arbitrary edge (u, v) on the network. Using a case analysis we show that this edge can always be tested.

- if u or v are beacons, then they can send a message directly on this edge,
- otherwise there exists a beacon b distinct from u and v; this beacon sends a message to u and v, if either of these messages traverses the edge (u, v) we are done,

- let $b, r_1, r_2, \ldots, r_k, u$ denote the path of a message from b to u. Now, b sends a message destined for v via r_1. Observe that since all nodes in the network offer transit and u is connected to v, the message for v must be delivered through this path. If the message traverses the edge (u, v) then we are done,

- otherwise, the message destined for v via r_1 shares a portion of the path from b to u, namely, $b, r_1, r_2, \ldots, r_j$ for some $1 \leq j \leq k$. This means that at the node r_j the paths bifurcate and v is reachable via both r_{j+1} and its default route. Hence r_i is of higher arity and is a beacon. Moreover, notice that r_i is now closer to u than b. Now by recursion, we can repeat the same case analysis, and as the edge-distance is reduced at some point one of the three earlier cases apply and the edge (u, v) is traversed by a message from a beacon.

□

Notice that the Claim above gives an effective —albeit perhaps not always cost efficient— method to deploy a beacon set. We can reduce the size of the beacon set as follows.

OBSERVATION 1. *Let $(u_0, u_1, \ldots u_k)$ be a path of high arity nodes in the network such that u_i is only connected to nodes u_{i-1} and u_{i+1} for $1 \leq i \leq k-1$. That is every message from a node in the interior of the path traverses to the outside network through either u_0 or u_k. Then the high arity nodes minus the set $\{u_1, \ldots, u_{k-1}\}$ is also a beacon set.*

Indeed, this is so as every message originated from a node in the interior of the path traverses through one of u_0 or u_k with the exception of a message destined to another node u_i in the path, but these edges can be tested by sending a message from u_0 to u_k along the link (u_0, u_1). Thus u_0 and u_k suffice for the beacon set and the nodes in the interior of the path can be omitted from the set.

This substantially reduces the size of the beacon set. The number of AS's providing transit on the Internet is in the order of 1500, as reported by the Asia Pacific Network Information Centre (APNIC) on 5 May 2001, [6]. Notice that large multihomed AS's are likely to have more than one beacon node, even after applying the observations above. For example, in the case of NSPs one would expect roughly one beacon per each peering point (public or private), plus other beacons for every cycle in the network. A quick glance at publicly available maps of some of the major backbones[2] show that most cycles in NSP networks contain at least one peering point, provided we treat multiple *direct* links between two points as a single bundle. In other words, the Internet is mostly a tree, except for public peering points and very short redundant paths such as FDDI rings and $n \times m$ fabric at PoPs or between core routers and border routers[3]. Therefore we can further reduce the size of the beacon set to approximately those nodes in the peering points.

Placing a beacon on each peering point and border router of a multihomed AS is likely to be a good approximation of a beacon set.

A set of routing data collected at Internap, as well as statistics released by APNIC suggest that the total number of multihomed networks is in the order of 10,000 to 20,000. Indeed a recent IETF internet draft (work in progress) [34] suggests capping the total number of mulithomed networks at 2^{15} or approximately 32,000. Hence we can expect that, anywhere between $1,500$ and $20,000$ beacon nodes suffice to cover the entire network. This number, while large, is much smaller than the total number of hosts, estimated at 171 million as of January 2003, [24, 33], and well within the economic reach of a large commercial Internet organization.

[2] We studied AT&T, Intermedia, GTE and UUNET.

[3] Others have noted that the "almost a tree" nature of the Internet makes some otherwise difficult or intractable problems tractable. In particular Xiao and Ni point out that OSPF can be extended to much larger organizations if proper note is made of tree like regions, which they term WARR's [38].

6. HIGH ARITY NODES

Thus far we have focused on the role of high arity nodes as part of the infrastructure required to measure connectivity and, by extension, performance path characteristics on a network such as the Internet. However because of their strategic placement a beacon set plays also a key role in realizing a Resilient Overlay Network (RON) [5] with a performance based routing policy.

The BGP protocol admits aggregation of paths which considerably reduces the size of the routing tables. On the flip side the lack of explicit performance characteristics means that the path chosen by BGP is not necessarily optimal latency-wise. This is further compounded by deviations from the AS-hop metric due to other considerations such as redundancy, cost-of-bandwidth and even lack of visibility into the performance characteristics of the network. Nevertheless, latency and packet loss are often driving characteristics of user bandwidth requirements [28].

Some commercial organizations provide some level of performance improvements on the public Internet over the standard BGP routing heuristics using network route optimization (e.g. Internap, Sockeye). These improvements are partially constrained by the imprecise granularity of BGP routing policy and lack of control across the network. Alternatively it is possible to deploy a performance based routing protocol network overlaid on the public Internet using tunnelling across strategically placed nodes in the network. Since the high arity nodes have access to all paths it is possible to increase granularity of routing decisions by placing forwarding-tunnel router nodes on that set.

A set of forwarding-tunnel router nodes is said to be an *all-paths set* if every simple path from a node u to v can be realized with it.

CLAIM 5. *The set of nodes of higher arity in a network form an all-paths set.*

PROOF. Recall that, from the proof of Claim 4, we know that all bifurcation points on the network are of higher arity. Hence all nodes where a routing policy can be implemented are part of the set of nodes of higher arity, and the nodes absent from the higher arity set are exactly those where paths are uniquely determined. Now consider the default path $RP(u, v) = u, u_1, \ldots, v$ from a node u to a node v, and an alternative, desired path $P(u, v) = u, w_1, w_2, w_3, \ldots, v$. Let w_i be the first node in which the two paths differ. Hence w_i is a node of higher arity, so we can send a message from u to w_i which by the monotonicity of the routing policy will follow the path $RP(u, v)$ up to node w_i. At this point since w_i is of higher arity and thus part of the forwarding-tunnel router nodes it can forward the message to v via w_{i+1}.

Now we repeat the above process with the $RP(w_{i+1}, v)$ and the path $P'(w_{i+1}, v) = w_{i+1}, w_{i+2}, \ldots, v$, determining the first node in which they differ, which, as before must also be a forwarding-tunnel router node. After each iteration i we obtain a routing path from u to a w_{j_i} with $j_i > j_{i-1}$. Hence after a finite number of steps this recursion must end and we have a set of forwarding-tunnel router nodes realizing the path $P(u, v)$. □

Notice that as in the case of the beacon set, Observation 1 can also be used to reduce the size of the high arity set while still maintaining the all-paths set property.

7. CONCLUSIONS

We have shown that computing the minimum number of sbeacons required to test the status of every link is NP-hard. This number is also hard to approximate and potentially as large as one-third of the nodes on an arbitrary network. An alternative heuristic tailored for the topology of the public Internet using high arity nodes is proposed. This would form a beacon set that can test for connectivity on all relevant edges of the network. Furthermore such a set has interesting properties that allow to further reduce the number of required nodes. The high arity set can also be used as a forward tunnelling set for all-paths routing on the public Internet, creating a QoS based RON.

8. REFERENCES

[1] A. Adams, T. Bu, R. Caceres, N. Duffield, T.Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley. The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior, *IEEE Comm.*, 2000.

[2] A. Adams, J. Mahdavi, M. Mathis, and V. Paxson, Creating a Scalable Architecture for Internet Measurement. *Proc. 8th Internet Society Conf. (INET),* 1998.

[3] A. Adams, and M. Mathis. A system for flexible network performance measurement. *Proc. 10th INET Conf.*, 2000.

[4] M. Adler, T. Bu, R. K. Sitaraman, D. F. Towsley. Tree Layout for Internal Network Characterizations in Multicast Networks. *Networked Group Comm.*, 2001, pp. 189-204.

[5] D. G. Andersen, H. Balakrishnan, M.F. Kaashoek, R. Morris. Resilient Overlay Networks. *Proc. 18th ACM Symp. on Operating Syst. Princ.*, 2001.

[6] Asia Pacific Network Information Centre (APNIC). *Daily BGP statistics.* `http://www.apnic.net/stats/bgp`. May 5, 2001.

[7] Cooperative Association for Internet Data Analysis (CAIDA). *The Skitter Project.* `http://www.caida.org/tools/measurement/skitter/index.html`, 2001.

[8] P. Barford, A. Bestavros, J. W. Byers, M. Crovella. On the marginal utility of network topology measurements. *Internet Measurement Workshop*, 2001, pp. 5-17.

[9] O. Bonaventure, S. De Cnodder, J. haas, B. Quoitin, R. White. Controlling the redistribution of BGP routes. Internet draft.

[10] S. Branigan, H. Burch, B. Cheswick, and F. Wojcik. What Can You Do with Traceroute? *Internet Computing*, vol. 5, no. 5, 2001, page 96ff.

[11] T. Bu, N. G. Duffield, F. Lo Presti, D. F. Towsley. Network tomography on general topologies. *ACM Int. Conf. on Measurements and Modeling of Comp. Systems (SIGMETRICS)* 2002, pp. 21-30

[12] R. Caceres, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network internal loss characteristics. *IEEE Transactions on Information Theory*, v.45, n.7, 1999, pp. 2462-2480.

[13] Bill Cheswick, Hal Burch, and Steve Branigan. Mapping and Visualizing the Internet. *Proc. USENIX Technical Conf.*, 2000.

[14] K. Claffy, G. Miller and K. Thompson. The nature of the beast: recent traffic measurements from an Internet backbone. *Proc. 8th Internet Soc. Conf. (INET),* 1998.

[15] K. Claffy, T.E. Monk and D. McRobb. Internet Tomography. *Nature*, 7th January 1999.

[16] X. Deng. Short Term Behaviour of Ping Measurements. *MSc thesis, Univ. of Waikato,* 1999.

[17] N. Feamster, J. Borkengham, J. Rexford. Controlling the Impact of BGP Policy Changes on IP Traffic. Technical Memorandum, AT&T Labs Research.

[18] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewicz, Y. Jin. An Architecture for a Global Internet Host Distance Estimation Service. *Proc. IEEE Conf. on Comp. Comm. (INFOCOM),* 1999, pp. 210-217

[19] M. Garey and D. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* W.H.Freeman, 1979.

[20] R. Govindan, H. Tangmunarunkit. Heuristics for Internet Map Discovery. *Proc. IEEE Conf. on Comp. Comm. (INFOCOM),* 2000, pp. 1371-1380

[21] I. D. Graham, S. F. Donelly, S. Martin, J. Martens and J. G. Cleary. Nonintrusive and accurate measurements of unidirectional delay and delay variation in the Internet. *Proc. 8th Internet Society Conf. (INET),* 1998.

[22] R. Gúerin and A. Orda. QoS-based routing in networks with inaccurate information. *Proc. IEEE Conf. on Comp. Comm.ons (INFOCOM),* 1997.

[23] B. Halabi. *Internet Routing Architectures.* New Riders Publishing, 1997.

[24] Internet Software Consortium. `http://www.isc.org/ds/WWW-200301/index.html`.

[25] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang. On the Placement of Internet Instrumentation. *IEEE Conf. on Comp. Comm. (INFOCOM)*, 2000, pp. 295-304.

[26] S. Kalidindi and M. J. Zekauskas. Surveyor: An infrastructure for Internet performance measurements. *Proc. 9th Internet Soc. Conf. (INET),* 1999.

[27] G.R. Malan and F. Jahanian. An extensible probe architecture for network protocol performance measurement. *Proc. ACM Conf. on Applications, Technologies Architectures and Protocols for Comp. Comm.* (SIGCOMM), 1998.

[28] A. Odlyzko. The current state and likely evolution of the Internet *Proc. Globecom'99, IEEE*, pp. 1869-1875, 1999.

[29] V. Paxson. Measurements and Analysis of End-to-End Internet Dynamics. *PhD thesis, Univ. of Cal., Berkeley,* 1997.

[30] V. Paxson. End-to-End routing behaviour in the Internet. *IEEE/ACM Transactions on Networking.* 5, 601-618 (1997).

[31] V. Paxson, J. Mahdavi, A. Adams and M. Mathis, An Architecture for Large-Scale Internet Measurement. *IEEE Comm.*, v.36, n.8, 1998, pp. 48-54.

[32] R. Raz and S. Safra. "A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP", *Proc. 29th ACM Symp. on the Theory of Computing (STOC)*, 475-484, (1997).

[33] A. Scherrer. *127,781,000 Internet Hosts: How Matrix.net gets its host counts.* `http://www.matrix.net/isr/library/how_matrix_gets_its_host_counts.html`, 2001, Access: May 2001.

[34] P. Savola. Multihoming using IPv6 addressing derived from AS numbers. draft-savola-multi6-asn-pi-00.txt, work in progress. IETF internet draft, January 2003.

[35] S. Seshan, M. Stemm, and R.H. Katz. SPAND: Share Passive Network Performance Discovery. *Proc. 1st Usenix Symp. on Internet Technologies and Systems*, 1997.

[36] R. Siamwalla, R. Sharma, and S. Keshav. Discovering Internet Topology. Technical Report, Cornell Univ., July 1998.

[37] D. Towsley. Network tomography through to end-to-end measurements. Abstract in *Proc. 3rd Workshop on Algorithm Engineering and Experiments (ALENEX),* 2001.

[38] X. Xiao and L. M. Ni. Reducing routing table computation cost in OSPF. *Proc. 9th Internet Society Conf. (INET),* 1999.