High Arity Nodes, Routing and Internet Tomography

J.D. Horton
Faculty of Computer Science
University of New Brunswick
Fredericton, N. B. E3B 5A3
Canada.
Email: jdh@unb.ca

A. López-Ortiz
Director of Core Research,
Internap Network Services, Seattle, WA
and
Fac. of Computer Science, UNB.
Email: alopezo@unb.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 measurement techniques using strategically placed nodes called beacons. We show that computing the minimum number of required beacons on a network under a BGP-like routing policy is **NP**-complete and at best $\Omega(\log n)$-approximable. In the worst case at least $n/4$ and at most $(n+1)/3$ beacons are required for a network with n nodes. We then introduce some results and 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 as well as interesting economic settlement properties for public peering.

1 Introduction and Motivation

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 [12]. Empirically, it has been observed that a few key failures often have significant impact on routing decisions. Using a mathematical model Albert et al. predict that the Internet could see severe effects with as little as 2.5% node failures [1].

Routing decisions and content distribution networks (web caches) require proper connectivity and latency information so as to direct traffic in an optimal fashion.

In this work we consider the problem of determining the topology of the Internet under the assumption that a few thousand beacons running special software are deployed at key sites across the entire Internet. Gathering connectivity information through beacons is known as Internet Tomography. Currently there are several efforts in progress to obtain topology and performance measurements on the Internet [15, 11, 3, 17], several of which use some form of beacons or agents to extract information from the network.

In practice these beacons are often placed in universities and other organizations that are willing to host the software or hardware required for these measurements. These beacons are placed in "key points" of the network according to various heuristics.

In this paper we study the optimal and systematic placement of these beacons and the properties of a beacon set mapping the network. We show that computing the minimum number of required beacons is **NP**-complete under a BGP-like routing policy. Using a reduction to minimum set cover we prove that at best this problem is $\Omega(\log n)$-approximable. We also provide a lower bound in terms of the number of nodes on the network. In the worst case at least $n/4$ and at most $(n+1)/3$ beacons are required for a network with n nodes.

We then show that placing beacons on high arity nodes suffices to map the Internet. Moreover, it follows from the analysis that by placing special tunnelling nodes on high 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. This has important implications for guaranteed quality of service (QoS) bandwidth providers. By their nature these tunnelling nodes bring economic settlement to peering points as well as "byte-mile" economic settlement to the public Internet.

2 Definitions

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

We model the network as a graph. Hosts correspond to nodes and links to edges. Every node in the network can apply local routing policy decisions. However, as stated above, those routing policies are such that the network is fully 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 a non-negative weight indicating some metric such as latency or AS hop distance. The path taken by a message is known at the source. In particular, in the case of the Internet this can be obtained separately through a `traceroute` call.

We assume that global connectivity information is dispersed in a BGP-like fashion. This protocol transmits connectivity information in rounds. Each node sends to its neighbours a set of nodes that are known to be reachable from it together with a cost attached to that connectivity. With this information each node updates its local connectivity information to reflect those nodes reachable through a neighbour and broadcasts updates in the table to its neighbours. This process is repeated until no further updates are transmitted.

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.

Definition 1 *A node n of degree k in the network is said to have* arity m *for $m < k$ if it can send a message through m different edges to at least one node v.*

Notice that as the network is fully connected every node can be reached through at least one path, therefore every node has arity of at least one.

Definition 2 *A node with arity $m \geq 2$ is said to have* high arity*.*

On the Internet, a collection of nodes under a single routing policy and running under a single technical administration is called an AS [10]. An AS is said to be multihomed if it has more than one connection to the cloud. Alternatively, under our terminology an AS is multihomed if it contains at least one border node with high arity to a node external to the AS.

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. In the latter case the message is sent back towards the node from whence it came. Thus a depth first search occurs when sending a message. Hence messages always will be delivered if the network remains connected.

Definition 3 *Let v be a node reachable from a high arity node u through two different edges (u, q_0) and (u, q_1). Then u is said to offer transit if node v can be reached by sending a message from q_0 to u destined to v and from q_1 to u destined to v.*

High arity nodes necessitate a routing policy. 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). Only those nodes having an arity higher than one require knowledge of a routing policy to determine in which of several valid directions to forward a message.

Definition 4 *Let $(u_0, u_1, \ldots, u_n)$ be the path taken by a message from node u_0 to node u_n using standard routing policy. Then the network is said to have monotonic routing if the path under standard routing policy 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.*

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.

3 The Beacon Placement Problem

First we consider a network in which no new links are added. A given link might become unavailable but otherwise routing policy remains consistent with the weights given. 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. We assume that at all times the network remains connected, that is, no failure is such that breaks the network into two components.

Definition 5 *The Beacon Placement Problem is to determine the minimum number (and/or position) of beacons on a network of known topology so that for every edge in the network there exists a sequence of messages originating from nodes of the beacon set that can determine the edge status, regardless of other failures, as long as the network remains fully connected.*

This assumes that the routing policy remains constant. In other words save for edge failures the high arity node set remains unchanged.

Definition 6 *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 transmition fails, then such a placement of beacons is called a* beacon set*.*

Theorem 1 *A necessary and sufficient condition for a collection of beacon nodes to determine if any 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 an 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.

Conversely assume by way of contradiction that we have a set of beacons which cannot transit an edge. Then if that edge goes down, the beacon set cannot detect this fact as the edge is not transited which is a contradiction. □

BGP supports a variety of mechanisms to establish routing policy. In practice, AS-hop path length heuristic and administrator defined preferences are two of the most common. We consider first a network in which all policy decisions are made based on local preferences. In this case, in each high arity node, the administrator declares a single preferred NSP whenever more than one choice is available.

Definition 7 *Given two nodes n and m, define the* maximal initial shortest path from n to m, *MISP*(n, m)*, to be the path followed by messages sent from n to m. Define the* maximal initial shortest path spanning tree of a node n, *MISPST*(n) *to be the union of the MISP*(n, m) *over all possible nodes m.*

The *MISPST*(n) for a given computer n can be found by doing a breadth first search from n, sorting the edges at each node in decreasing capacity.

```
Algorithm MISPST($n$)
Input:  A graph
Output: A tree
$T \leftarrow$ empty tree
$Q \leftarrow$ new Queue
push $n$ onto $Q$
while $Q$ is non-empty do
        $m \leftarrow$ pop $Q$
        sort the outgoing edges at $m$ by decreasing capacity
        for each outgoing edge $(m, x)$ by decreasing capacity
            if $x$ has not been on $Q$
               then
                      push $x$ onto $Q$
                      put edge $(m, x)$ into $T$
        end for;
end while;
return $T$
```

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 *MISP*(n) is down, assuming that no other edge is down. The node n sends messages to all nodes in the *MISP*(n) in a bread-first search and compares the path used against the *MISP*(n). A difference in the path indicates a failed edge, by sending a message to y. If (x, y) is not down, and no other edge on *MISP*(n, y) is down, then the

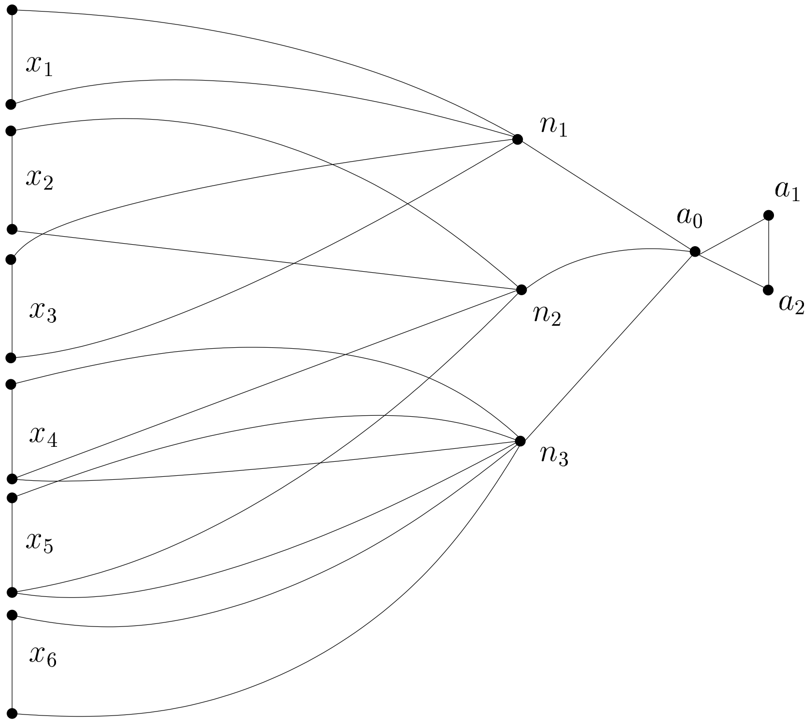

Figure 1: Set covering reduction to beacon placement.

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 *MISP*(n, y) is down, then we can find this out, and by the assumption that at most on edge is down, we then know that the edge (x, y) is not down. If (x, y) is down, then sending the message from n to y may or may not get through. If the message gets through, then the message must follow *MISP*(n, x) and then not take the edge (x, y) which it is supposed to.

Lemma 1 *The edges that a node n can determine by polling to be not down consists of all the edges of the $MISPST(n)$ and also for each neighbour m of n, the $MISPST(m)$.*

Proof. The node n can send messages through all the edges of $MISPST(n)$, and also for any neighbour m of n, through all the edges of $MISPST(m)$. The latter trees can be searched because n as a beacon, can send m a message with any other node in the network as destination. There are no other edges that n can send a message through, so these are all the edges that n can detect as down. □

There exists a reduction of Set Cover, which is known to be **NP**-complete [7], to the Polling Problem.

Theorem 2 *The Polling Problem is NP-Complete.*

Proof. We prove this by transformation from Minimum Set Cover. In this problem an instance is a collection of sets $S_1, S_2, \ldots, S_m \subset S = \{x_1, \ldots, x_n\}$ and the objective is to obtain a subcollection S' of sets S_i such that jointly they contain S, i.e. $\cup_{S_i \in S'} S_i = S$.

For each $x_i \in S$, define an edge $e_i = \{v_i, u_i\}$ of capacity 1 . For each S_j, define a computer n_j. 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 capacity 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 1, 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 length preference using weights as AS hop distance. 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 corresponding to a set S_j containing x_i. 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 $n_0 \cup \{n_j \mid S_j \in S'\}$ is a polling set if and only if S' is a set cover of S. □

Corollary 1 *The Polling Problem has no approximation algorithm with a performance ratio better than $\Omega(\log n)$.*

Proof. Note that the transformation maps each set to a distinct polling computer. Moreover, as shown by Raz and Safra [14] there is no approximation algorithm for Minimum Set Cover with a performance ratio better than $c \log n$ for some positive constant c. □

Theorem 3 *A network with n nodes may require up to $n/4$ polling stations to determine whether an edge is down or not.*

Proof. Consider the network shown in Figure 2. Every vertical edge in each diamond can only be polled by nodes within its own diamond. This means there is a scout for every four-node diamond. □

Theorem 4 *Any connected network of n computers has a set of not more than $(n+1)/3$ computers which form a polling network.*

Proof. Build a depth-first-search spanning tree using unweighted link distance metric of the network. Colour every node in this DFS-tree by its distance to the root modulo 3. This partitions the set of nodes into three classes, the 0-class the 1-class and the 2-class. By convention the root is in the 0-class and the 2-class. Every edge in the tree is distance at most two of some vertex in each class, and hence can be polled by that node. Therefore, each of these three classes is a polling set. Since the sum of the cardinality of their unions is $n+1$, the smallest of these classes has cardinality less than $(n+1)/3$. □

Theorem 5 *Placing a beacon on every high arity node forms a beacon set.*

Proof. This follows from Lemma 1 and the definition of high arity. Consider a given edge, if the end nodes can only be reached in one way from a beacon then it is part of the MISP of that beacon and therefore the edge can be tested. If there are multiple paths to the end nodes, the beacon set contains nodes in all forks, leading to the edge and therefore the edge can be tested. □

Notice that Theorem 5 above gives an effective —albeit perhaps not always efficient— method to deploy a beacon set. We can reduce somewhat the size of the beacon set as follows.

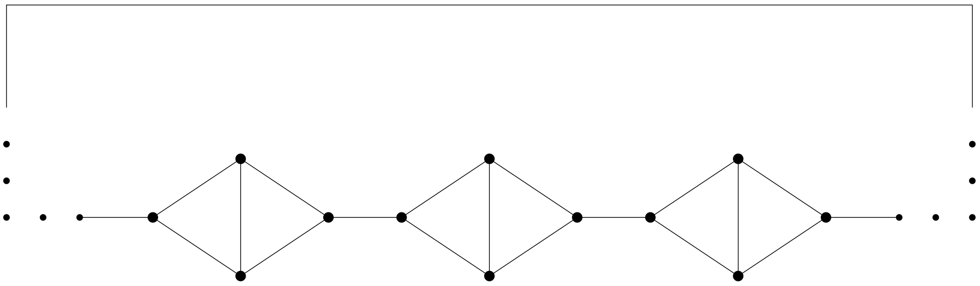

Figure 2: A diamond necklace.

Lemma 2 *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.*

Proof. 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. □

The Lemma above reduces substantially the size of the beacon set, which together with the following observation bound further the size of the beacon set required.

Empirical Observation 1 *The number of ASes providing transit on the Internet is in the order of 1500.*

This information is as reported by the Asia Pacific Network Information Centre (APNIC) on 5 May 2001. The data is derived from BGP IPv4 routing tables in an APNIC router located in Japan [2].

Notice that large multihomed ASes are likely to have more than one beacon node, even after applying Lemma 2. 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[1] 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 between core routers and border routers[2]. Therefore using Lemma 2 and this observation we can restrict the beacon set to approximately those nodes in the peering points.

Empirical Observation 2 *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.*

Using Empirical Observations 1 and 2 we conclude that the total number of beacons required to map edges of the public Internet is likely to be in the range of $1,500$ to $10,000$

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

[2] 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 [18].

nodes. This number, while large, is much smaller than the total number of hosts, estimated at 128 million [15], and well within the economic reach of a large commercial Internet organization

4 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 performance based routing.

BGP uses an AS-hop metric augmented with a set of local preferences set by the network administrator at the local level. The AS-hop metric is a heuristic that 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 this policy 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 [16]. NSPs can only guarantee latency based routing policies within their network which are guaranteed across their backbone through Service Level Agreements (SLA), but once a packet transits over a peering point to a different network performance is no longer guaranteed.

Some commercial organizations have arisen providing some level of performance improvements on the public Internet over the standard BGP routing heuristics (e.g. Internap, Optnix). These improvements are partially constrained by the imprecise granularity of BGP routing policy, lack of visibility of performance characteristics and control across the network, although it must be said that the specific nature of their routing policies is proprietary and thus the precise impact of these problems on their routing mechanisms is unknown. A system based on MPLS, which has finer routing granularity, can be used to this end, but it will have to wait for wider support and deployment of this protocol.

Alternatively it is possible to deploy a performance based routing protocol 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.

Definition 8 *A set of forwarding-tunnel router nodes is said to be an* all-paths set *if every possible path from a node u to v can be realized with it.*

Theorem 6 *The high arity nodes in a network form an all-paths set.*

Proof. This follows from the definition of a high arity node. Any point where a path alternative might arise is a high arity node and thus all-paths can be taken as long as the high arity node can be used to tunnel a message across. □

Notice that as in the case of the beacon set, Lemma 2 provides a method to reduce the size of the high arity set while still maintaining the all-paths set property, so in fact this Theorem applies to this smaller set as well.

Interestingly, as all peering points are high arity nodes then an all-paths set brings economic settlement to peering points (public or private). That is, if all-path nodes are owned by an independent third party, standard commercial practices require economic compensation for each connection of the all-path nodes. Then as traffic is tunnelled from one NSP to another compensation would be extracted from each side. This is presumably a desirable characteristic of a network.

On the other hand, it is possible that for performance reasons a message with origin and destination inside a single NSP might travel along a lower latency route by traversing a shortcut through another NSP backbone for a portion of the path. Under traditional settlement schemes using an all-paths set, the NSP containing the origin and destination would be compensated twice, once when the message is first injected into the NSP, and second when the message reenters the original NSP network from the shortcut backbone. In this example the cost per byte is increased three-fold. Similarly a multi-AS path would also necessitate compensation to each of the AS transited, which would make the all-paths method economically unfeasible.

To this end a per-mile compensation based scheme would have to be implemented. Alternatively, this could also be achieved using dedicated point to point frame circuits which are commercially available from most NSPs. This would allow a private network with proprietary protocols to be overlaid on top of the Internet relying on TCP/IP on the public internet for the last AS-hop portion of the path.

In practice, this method would require NSPs to share the location of peering points with the provider of the all-paths service. In the past, NSPs have proved somewhat receptive to such special requests as they provide increased compensation for traffic. The extra cost is borne by the user which in exchange obtains increased quality of service thus justifying the increased cost of connectivity, if any (e.g. Akamai, Internap).

5 Conclusions

We have shown that, while valuable information can be gathered from strategically placed beacons in the Internet, computing the minimum number of such beacons is NP-hard. This number is also hard to approximate and potentially as large as one-fourth 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 connectivy and latency performance characteristics 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, introducing economic compensation to peering.

References

[1] R. Albert, H. Jeong and A-L. Barabasi. Error and attack tolerance of complex networks. Nature, 406:378-382 (2000).

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

[3] Cooperative Association for Internet Data Analysis (Caida). *Skitter* http://www.caida.org/tools/measurement/skitter/index.html. May, 2001.

[4] K. Claffy, G. Miller and K. Thompson. The nature of the beast: recent traffic measurements from an Internet backbone. *Proceedings of the 8th Annual Internet Society Conference (INET),* ISOC, 1998.

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

[6] X. Deng. Short Term Behaviour of Ping Measurements. *MSc thesis, University of Waikato,* July, 1999.

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

[8] 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. *Proceedings of the 8th Annual Internet Society Conference (INET),* ISOC, 1998.

[9] R. Gúerin and A. Orda. QoS-based routing in networks with inaccurate information. *Proceedings of the IEEE INFOCOM'97,* 1997.

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

[11] S. Kalidindi and M. J. Zekauskas. Surveyor: An infrastructure for Internet performance measurements. *Proceedings of the 9th Annual Internet Society Conference (INET),* ISOC, 1999.

[12] V. Paxson. Measurements and Analysis of End-to-End Internet Dynamics. *PhD thesis, University of California, Berkeley,* April 1997.

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

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

[15] 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, last access May 4, 2001.

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

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

[18] X. Xiao and L. M. Ni. Reducing routing table computation cost in OSPF. *Proceedings of the 9th Annual Internet Society Conference (INET),* ISOC, 1999.