Algorithmic Foundations of the Internet Roundup*

Alejandro López-Ortiz

School of Computer Science
University of Waterloo
Waterloo, ON, Canada
email: `alopez-o@uwaterloo.ca`

Abstract. In this paper we present a short overview of selected topics in the field of Algorithmic Foundations of the Internet, which is a new area within theoretical computer science.

1 Structure of the Roundup

This paper presents a roundup of subjects not covered in the comprehensive surveys before it. These topics are also representative of the field of algorithmic foundations of the Internet. For each topic covered in the survey we start by describing at a high level what the problem or object of study is. We then give a brief historical perspective of how the challenge posed has been addressed so far within the network community, whenever relevant. Lastly we describe the theoretical and algorithmic aspects of the topic or challenge in question.

2 Web Caching

The world wide web was created in 1991, and became wildly popular in 1993 after the release of the graphics friendly Mosaic browser. Shortly thereafter it became clear that HTTP transactions would come to be the dominant consumer of bandwidth on the Internet [38]. Indeed this became the case sometime around 1995 and remained so until 2002, when peer-to-peer music and video sharing overtook HTTP traffic by volume. Even today it remains the second largest user of bandwidth, by protocol.

Being a large consumer of bandwidth means that reductions in unnecessary web retransmissions could potentially have a significant impact on the bandwidth requirements of an organization.

The term caching denotes the management of data being swapped between various storage media with differing transfer rates, and in particular, most often it is used to describe the transfer of data files or virtual memory between RAM and hard disks. For example, in this classical sense the operating system

---

* An earlier version of this survey appearead in "Algorithmic Foundations of the Internet", A. López-Ortiz, SIGACT News, v. 36, no. 2, June 2005.

is responsible for paging-in the working set of a program, while the rest of the instruction data might reside in disk. Accessing data across the Internet, such as a web page, also involves differential rate accesses as in the classical setting. Hence it is only natural that web traffic is amenable to caching speedups. Surprisingly, the original data transfer protocols such as FTP and HTTP did not initially incorporate notions of caching in their operations. Yet the need for them is the same, if not greater.

In Figure 1 we illustrate the typical path of a web page request. Naturally, caching might occur at any node along the path. Furthermore, if we take advantage of some HTML features it is possible to cache data off-path, using a cache farm.

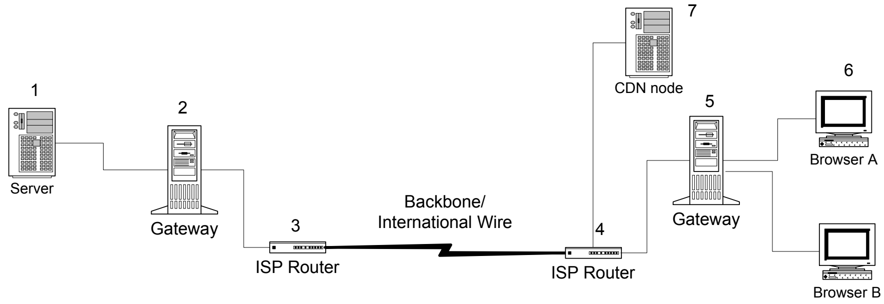

Fig. 1. Web page access path over the network.

2.1 Web Caching in Practice

Efforts to develop a web caching infrastructure started as soon as it became clear that web traffic growth was unprecedented. The first organizational proxy cache was introduced by DEC in 1994. A proxy cache serves as a centralized store for all HTTP requests generated from within an organization. The proxy mediates between the faster internal LAN and the slower external Internet connection. This corresponds to node 5 in Figure 1. In this way if client A requests a web page and client B issues a second request for the same web page within a short period of time, the second request can be served from the proxy storage without causing further traffic on the external network.

Similarly, the Netscape browser introduced a hard-drive mapped file cache (see node 6 in Figure 1). This cache, which is now standard in all browsers, was designed with the primary consideration of speeding up the "back" operation in the browser. As the user backtracks on a sequence of links a local copy of a previously viewed file is served instead of requesting a new copy across the network. As trivial as this might seem such a local cache system was lacking in the Mosaic browser as well as in most access clients for the other two major file transfer protocols at the time, namely, FTP and gopher.

A national cache hierarchy was implemented in New Zealand in 1994. This corresponds to node 4 in Figure 1. A cache hierarchy is a caching infrastructure serving more than one organization or ISP (Internet Service Provider). Since the cost of international traffic over an undersea cable to New Zealand was high, great savings could be derived from pooling resources to create a national cache hierarchy.

Other forms of caching were slower in coming. Network-edge proxies, which cache content from the server to the network edge, (corresponding to node 2 in Figure 1) first appeared in the year 2000 or so. This type of caching reduces traffic on the internal network where the server is located. Since the internal network is typically used for business critical functions, while web servers rarely are, a network edge cache realigns the use of internal bandwidth with local parameters. Secondly, a network edge cache precludes external HTTP packets from traversing in the internal organizational network. Such external packets could potentially become a security risk, thus reducing or eliminating them is considered a desirable side effect of network edge caching.

Similarly geographic push caching was first proposed in [18], and subsequently supported by experimental setups [31]. Yet, surprisingly such caching schemes did not become popular until the year 2000 with the appearance of widely deployed content distribution networks (CDN) such as Akamai and Digital Island. In this type of caching the server proactively disseminates content to caching nodes located across the network. When the client requests a web page, its request is redirected to a geographically close cache server by a combination of DNS redirection and secondary content redirection, that is, redirection of `<IMG...>` links (see node 7 in Figure 1).

To this date, we are not aware of a caching solution that is deployed at the national level on the server side (node 3 in Figure 1). Similarly we do not know of any web specific caching algorithms at the web server: any caching that takes place in it is derived from the file cache management in the operating system.

2.2 Algorithmic and Theoretical Aspects

In terms of content networks, theoretical efforts have taken place to create optimal schemes for load balancing and content redirection in CDNs [27]. However these issues are secondary parameters in the efficiency of CDNs, as content location is preeminently mandated by geographic considerations rather than load balancing issues.

The problem of determining optimal geographic location for cache placement has been the subject of intense theoretical study. Consider first the extension of Figure 1 to include all nodes that access a specific web page, e.g., all users accessing the SIGACT home page as shown in Figure 2. The result is a graph with the SIGACT server as root and the computers running web browsers (clients) as nodes of degree 1 (see Figure 2). In general such a graph might have an arbitrary topology although it has been observed in practice that the graph obtained is very nearly a tree, in the sense that it has n nodes and $n+c$ edges, for a constant $c \ll n$. Consider now a CDN company which wishes to place k-caching servers

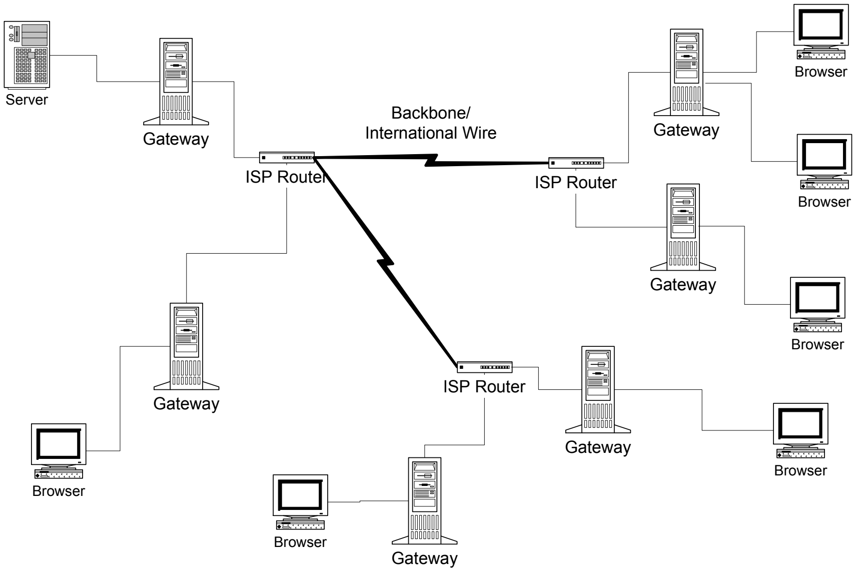

Fig. 2. Network of distribution paths for a single server.

on the tree nodes in such a way as to minimize the total amount of traffic used in serving the requests of the clients. The working assumption is that a request is served by the first cache node appearing in the path from the client to the server. Computing the optimal location of the caching servers has been shown to be NP-complete by [28] for arbitrary topologies of the graph, but it is known to be computable for tree topologies [30]. Other variants of this problem have been considered, with differing assumptions of the model. In general, this problem is closely related to the well known p-median server problem, which is NP-complete. Hence many of the variants studied remain NP-complete under a general network topology or even assuming a more restricted Internet-like topology (see e.g. [24]).

Another important parameter in a CDN is the cache management strategy. That is, which files are replicated where and how are they evicted. Each server in a farm can be treated as an on-line isolated object store in which case the server may use classical standard caching strategies such as LRU-like policies (more on this later), or the entire set of k servers can act in a coordinated way, establishing a cache policy that also reflects centrally collected historical access patterns. These globally coordinated strategies are known as *replica placement algorithms* [24]. A general survey of the different algorithms can be found in [26] and an intriguing argument questioning the benefits of a coordinated policy over single server strategies appears in [25].

Web cache management strategies were first considered by Markatos who observed that web files are served on an all-or-nothing basis, since in most cases users either request the entire page or nothing at all [34]. This is in contrast to the classical setting, where, for example, access to data using a database does not result in loading the entire database file, but only the relevant portions of the index structure and the requested datum itself. Markatos observed that this difference was significant in terms of cache performance and gave empirical data to support the observation that standard RAM-to-hard-drive cache management strategies were unsuited for web servers. This all-or-nothing characteristic results in a cache policy in which objects cached are of different sizes. Aside from this fact, a web cache differs from a RAM cache in other ways. First, a web page miss on a full cache does not necessarily result in a page eviction from cache as it is possible to serve the file straight from disk to the network card bypassing the cache. Second, there is a dual latency parameter in that there is lag time to load the file from hard drive to memory and a second lag in serving the file across the network. This means that in practice when serving a large file, the latency of transmission over the network dominates the transfer time from external storage to memory. Hence a cache fault on a large file on hard drive has no impact on the latency perceived by web users. Third, the web server has access to extensive frequency of access statistics for the web files in question. In this way, the access logs can be used to compute an off-line or on-line cache management strategy that best reflects the access patterns to the web site. Lastly as the web objects are larger and accessed less frequently than in OS caching, it is possible to implement cache management strategies that are more computationally expensive per given access.

Irani was the first to propose and analyze an algorithm specifically designed for web files that took in consideration both frequency of access and multi-size pages [20, 9]. Interestingly, it was observed that the off-line optimum strategy of always evicting the page whose next request is furthest in the future—also known as Belady's rule—does not hold for multi-sized pages. Indeed, the offline optimum is NP-hard to compute under most models considered [4]. Another consequence of Belady's rule not holding is that the standard LRU-based on-line strategies used in the classic setting are not optimal for multi-sized pages. Indeed, the best known strategy for this problem has competitive ratio of $\Theta(\log n)$ in contrast to the constant competitive ratio of LRU in the classical server setting [20]. Several refinements of Irani's model have been proposed in an attempt to better reflect the costs of a cache miss in the web setting. Another aspect that has been incorporated is the second order correlations in the access patterns of web pages [11, 17]. Further details on web caching can be found in [21].

Aside from the benefits of reduced latency for the user (if there is a cache hit), web caching also results in increased processing capacity for the server (if the file can be served from the server's main memory), reduced server load (if the file can be served from a remote cache) and reduced demands on network bandwidth. Interestingly, this last not only benefits the client involved in the page request, but also other clients which will now encounter reduced network contention as a result of bandwidth savings from other users.

3 Internet economics and game theory

The Internet has a decentralized structure with no overarching commanding authority. At the same time, this is not to say that anarchy reigns on the Internet. To the contrary, the internet protocols fully specify the context and rules under which network interactions are to take place. On the other hand, if a specific feasible solution is not explicitly predicated by the protocols, there is no central authority to impose it. For example, consider a simple network as illustrated in Figure 3. In this case node A wishes to send a unit size message to node B while node C wishes to send a message to node D. We assume the capacity of all edges to be the same. A coordinating mechanism is necessary to ensure that only one of A or C chooses to send the message through edge e_1, while the other should choose edge e_2. The Internet does not provide a mechanism for this type of centralized coordinating action.

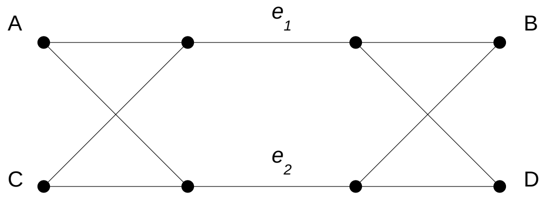

Fig. 3. Simple network.

In practice what happens is that each organization in the network makes its choices separately and mostly independently. The quality of each user choice, however, is dependent on the choices of all other users. Hence it is to one's own benefit to consider what are the likely choices of other organizations and incorporate those into one's own reasoning. This is exactly the type of reasoning that takes place in a classical game setting, such as chess, monopoly or poker: there is a set of commonly agreed upon rules, no central coordinating authority and self-interest is the main motivator.

A natural question is what game is defined by the the current Internet. That is, what are the individual payoff functions for players, where do random events take place and what exactly are the rules of the game.

Alternatively, the converse question holds. Consider a given strategy for selecting state, such as the TCP/IP congestion control protocol. What is the payoff function that is maximized by this protocol as currently defined? It could well be the case that the current function being optimized is not desirable to anyone in particular in which case a change in the protocol would be called for.

To illustrate further, consider a simple Internet game in which the players are web servers, serving content across the network. Making a move consists of choosing an injection rate of packets into the network. If the injection rate chosen is too high, then some of the packets sent are destroyed en-route to the destination and must be retransmitted, thus in the end reducing the effective transmission rate. If on the contrary the packet injection rate is too low, the server could increase its utility function value by increasing its injection rate. Observe that the injection rate that can be sustained is dependent on the amount of traffic being sent over any given link by all players. The goal for each player is to maximize its own effective transmission rate. Clearly this defines a game in the classical game theory sense.

In this injection game, the players aim to adjust their injection rate until it cannot be improved any further, that is, if any one of them were to increase or decrease its current injection rate this would result in a lower effective transmission rate for that player. That is to say, the players aim to reach a Nash equilibrium point. This is defined as a game state in which no single player has an incentive to alter its current choice of play. Formally, an n player game is a finite set of actions S_i for player i and payoff or utility functions $u_i : S_1 \times S_2 \times \ldots \times S_n \rightarrow \mathcal{R}$ for $i = 1 \ldots n$. A *(pure) Nash equilibrium* is a state of the game $(s_1, \ldots, s_n) \in S_1 \times S_2 \times \ldots \times S_n$ such that $u_i(s_1, \ldots, s_i, \ldots, s_n) \geq u_i(s_1, \ldots, s_i', \ldots, s_n)$ for all $i = 1 \ldots n$.

Currently the Internet uses a certain set of mechanisms (some by virtue of the protocol, others by common practices) to deal with injection rates that are too high. These mechanisms define the "rules" and hence the valid actions in the injection game above. A natural question to ask is if the rules as defined lead to optimal usage of the network. For example, it might be the case that the rules are overly cautious (e.g. TCP window size) and lead the players to select injection rates that are below what could be sustained over the current network topology. A formal analysis of the game should lead to the expected transmission rate that the players will reach. In turn this value can be compared with the maximum capacity of the network, as determined by standard network flow analysis.

A no less important question is if there exists *any* enforceable set of rules that could lead to an optimal usage of the network as deployed. In general it is known that there are games in which the Nash equilibrium points give suboptimal global behaviour, that is, in certain games the Nash equilibrium points result in outcomes undesirable to all parties (e.g. prisoner's dilemma, tragedy of the commons). These tend to arise in situations in which in the globally "best" solution there exists at least one player who can improve its lot by acting selfishly. The other players then must assume beforehand that such player will defect if given the opportunity and hence avoid such (otherwise preferable) states altogether. In other words, players actively avoid unstable states and aim for a Nash equilibrium in which players have no incentive to defect.

In this case it is important to determine how much worse is the outcome in the uncoordinated game setting as compared to a centrally mandated solution. For example, in the tragedy of the commons game scenario, there exists a shared grass meadow which is used for grazing. The globally optimal solution is for each of the n players to consume $1/n$ of the meadow at a sustainable rate. However, in such situation, a selfish player that consumes more than its fair share improves its utility function and hence has an incentive to cheat. This undesirable outcome can be avoided if we allow a central authority (typically the State, but not always necessarily so) to enforce a maximum rate of consumption of $1/n$. In the absence of such authority the optimal Nash equilibrium strategy in this case is for each player to consume as much of the meadow grass as possible, as fast as possible, until only the bare land is left (as actually happened in practice). In this last scenario all players end up being worse off.

The difference between the benefit players would have derived in the centrally planned solution and the Nash equilibrium solution is sometimes referred to as *the price of anarchy.*[1] For example, consider a routing game which is in some ways similar to the injection game. The players are nodes in the network, which can select one of several routes to send a message. The utility function for all players is to minimize the latency (travel time) of their individual messages. As in the injection game, the effective latency is affected by which routes are chosen by the other players. This game has a Nash equilibrium point as solution which we can compare with the globally optimal solution. The globally optimal solution is defined as that which maximizes the social welfare. In the routing game the social welfare is defined as the inverse of the sum of all latencies and the globally optimal solution is hence that which minimizes this sum of latencies[2].

Interestingly, it has been shown that the price of anarchy for the routing game is at most two in the worst case [37]. This is generally considered an acceptable penalty in exchange for the benefits of a globally distributed solution with no single point of failure and no need for active enforcement as players have no incentive to cheat.

Conversely, we could aim to modify the rules in such a way that in the resulting game the Nash equilibrium solution coincides with the globally optimal solution. This is sometimes refered to as *inverse game theory.* In standard game theory we are given a game and the goal is to identify the Nash equilibrium points. In inverse game theory we are given the desired state that we wish to achieve, and we wish to design a game that has that desired state as a Nash equilibrium point and no other. See [42] for a more detailed introduction to the subject of Internet routing and game theory.

4 Tomography

Efficient routing and caching require accurate connectivity information of the Internet. Internet protocols, on the other hand, make this task difficult as routing

---

[1] Be warned that there is a bit of hyperbole in the term "anarchy" as used in this instance, as in all cases we assume that somehow the rules of the game are still being enforced. Indeed economists refer to this milder "anarchy" under consideration as *laissez faire.* The true anarchist, no-rules-barred optimal solution for a player in the injection game would be to unplug the other servers.

[2] Observe that the choice of social welfare is generally not unique. For example we could define the social welfare as the maximum over all latencies or the difference between the maximum and minimum latencies.

decisions are made locally and shared across organizations only in aggregate form.

During the early years of the Internet this did not pose much of a challenge as the backbone was administered by the NSF with a relatively open policy. However, since the NSF relinquished control of the Internet backbone in the early 1990's the Internet has not been centrally managed. As a consequence, the topology of the network became a black box. In fact, backbone operators (National Service Providers or NSPs for short) generally consider the low level details of their respective backbone configurations a commercial secret both for competitive and security reasons.

Hence the information of interest, be it latency, topology, or connectivity, has to be inferred from experimental measurements. The inference of network topology and other network properties through indirect measurement came to be known as *internet tomography* [13, 14, 41].

4.1 Tomography in practice

Internet protocols are generally designed to build a layer of abstraction above the underlying hardware. This is a great advantage in terms of interoperability: since not much is assumed from the physical network media, most network hardware protocols can be (and have been) seamlessly incorporated into the Internet. The flip side of this is that the layer of abstraction hides low level details which might be useful for measuring certain network parameters such as routing quality, stability and security. Fortunately, not all low level details are hidden. In particular two of the most commonly used tools for low level discovery are `ping` and `traceroute` or derivatives of them. `ping` allows a computer to measure the round trip time for a message on between two computers on the Internet. `traceroute` returns the path taken by a packet from source to destination. Another effective method for obtaining connectivity information is to examine a set of BGP routing tables on the Internet. The routing table at a given node on the Internet contains a high level view of the routes used from that point on to any other computer on the network.

Paxson initiated the study of network properties using data collected through external access points [35]. Currently there are several efforts in progress to obtain topology and performance measurements on the Internet, several of which use some form of measurement points or agents called *beacons*. Beacons inject traffic into the network to extract information from the reported path and latency of the traffic generated. In practice these measurement points (beacons) are often placed in universities and other organizations that are willing to host the software or hardware required. For example the National Internet Measurement Infrastructure (NIMI) [2, 36, 16] is a concerted effort to deploy general purpose beacons with particular focus in scalability and flexibility. Building on top of the NIMI effort is the Multicast-based Inference of Network-internal Characteristics (MINC) project which aims to measure performance characteristics of the network through the use of end-to-end measurements [8]. Other efforts along these lines are [23,40,41,39,16]. The location of these beacons is determined according to various heuristics [1,3,6,12].

4.2 Algorithmic and Theoretical Challenges in Tomography

As we have seen, there have been substantial efforts directed at the deployment and use of distributed measurement systems. A key question is, what are the properties necessary for such a system to provide accurate measurements? At least in principle, there could be properties that cannot be measured from end-to-end beacons or the measurements returned by the beacons could be skewed in some manner. This is not a purely academic concern: Chen et al. showed that in the case of the high level AS topology, some of the techniques in use provided biased samples [10].

To be more precise, the Internet is divided into high level organizations called Autonomous Systems (AS). Each of these autonomous systems is a grouping of thousands of computers. Any two given ASes have few to no direct links in between them. In this way one can create a high level map of the topology of the network by considering one node per AS and direct connections between them as edges. Some early measurements of the AS topology were derived from routing tables—a seemingly sensible measurement technique. Such measurements suggested a power law distribution of the node degrees in the AS map [15]. The study by Chen et al. measured the same topology using an alternate, more accurate technique and obtained an AS topology map of the Internet which did not evidence a power law degree distribution [10,32,33]. In principle the same could possibly be the case for end-to-end measurements derived from beacons. This question can be addressed both at the empirical and theoretical levels, as we shall see.

Jamin et al. proposed theoretical methods as well as ad hoc heuristics for computing the location of a set of beacons whose aim is to compute the distance maps on the network [22]. Recently, Barford et al. provided the first systematic experimental study to validate the empirical observation that a relatively small number of beacons is generally sufficient to obtain an accurate map of the network [5]. Moreover, they show that the marginal utility of adding active measurement points decreases rapidly with each node added. They also provide an intriguing theoretical model of the marginal utility of an additional repeated experiment in a sequence of statistical measurements. Bu et al. consider the problem of the effectiveness of tomography on networks of general topology [7]. 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 has strong theoretical underpinnings.

Horton et al. showed that determining the number and location of the minimum beacon set is NP-hard for general topologies and in fact it is not even approximable to a factor better than $\Omega(\log n)$ [19]. Worse still in some networks the minimum number of beacons needed is $(n-1)/3$. If we consider that the Internet has on the order of 285 million computers, a 95 million computer beacon set would be impractically large. Interestingly, the theoretical analysis suggested reasons why a much smaller beacon set would suffice on the current Internet topology. The paper gives an effective method to bound the number of beacons needed—at somewhere less than 20,000 nodes. Building upon these results Kumar et al. propose a refinement of the model which produces a robust beacon set of even smaller size [29].

5 Conclusions

The field of algorithmic foundations of the Internet has seen rapid growth over the last decade. Results in algorithmic foundations of the internet regularly appear in general theory conferences. At the same time there has also been an increase in the number of papers with a theoretical bent in what traditionally had been applied networks conferences. This new field has attracted mathematicians, physicists, combinatorists, management science/economists, and, naturally, computer scientists. The challenges are numerous and developments are often immediate applicability to the internet.

References

1. A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, and D. Towsley. The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior, *IEEE Communications*, May 2000.
2. A. Adams, J. Mahdavi, M. Mathis, and V. Paxson, Creating a Scalable Architecture for Internet Measurement. *Proceedings of the 8th Internet Society Conference (INET),* 1998.
3. M. Adler, T. Bu, R. K. Sitaraman, and D. F. Towsley. Tree Layout for Internal Network Characterizations in Multicast Networks. *Networked Group Communication*, 2001, pp. 189-204.
4. Susanne Albers, Sanjeev Arora, and Sanjeev Khanna. Page replacement for general caching problems. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms*, (SODA) 1999, pp. 31–40.
5. P. Barford, A. Bestavros, J. W. Byers, and M. Crovella. On the marginal utility of network topology measurements. *Internet Measurement Workshop*, 2001, pp. 5-17.
6. S. Branigan, H. Burch, B. Cheswick, and F. Wojcik. What Can You Do with Traceroute? *Internet Computing*, vol. 5, no. 5, 2001, page 96ff.
7. T. Bu, N. G. Duffield, F. Lo Presti, and D. F. Towsley. Network tomography on general topologies. *ACM Conference on Measurements and Modeling of Computer Systems (SIGMETRICS)* 2002, pp. 21-30
8. 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.
9. Pei Cao, and Sandy Irani. Cost-Aware WWW Proxy Caching Algorithms. In Proceedings of *USENIX Symposium on Internet Technologies and Systems*, 1997.
10. Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The Origin of Power-Laws in Internet Topologies Revisited. In Proceedings of *IEEE Conference on Computer Communications*, INFOCOM, 2002.

11. Ludmila Cherkasova, and Gianfranco Ciardo. Role of Aging, Frequency, and Size in Web Cache Replacement Policies. In Proceedings of *9th International Conference High-Performance Computing and Networking,* (HPCN Europe) 2001, pp. 114–123.
12. Bill Cheswick, Hal Burch, and Steve Branigan. Mapping and Visualizing the Internet. *Proc. USENIX Technical Conference*, 2000.
13. 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.
14. K. Claffy, T.E. Monk and D. McRobb. Internet Tomography. *Nature*, 7th January 1999.
15. Michalis Faloutsos, Petros Faloutsos, Christos Faloutsos. On Power-law Relationships of the Internet Topology. In Proceedings of *ACM Conference on Applications, Technologies Architectures and Protocols for Computer Communication* (SIGCOMM), 1999. pp. 251–262.
16. P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewicz, Y. Jin. An Architecture for a Global Internet Host Distance Estimation Service. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM),* 1999, pp. 210-217
17. Alexander Golynski, Alejandro López-Ortiz, and Ray Sweidan. Exploiting Statistics of Web Traces to Improve Caching Algorithms. *Technical Report CS-2003-34, School of Computer Science, University of Waterloo*, 2003.
18. J. Gwertzman, and M. Seltzer. The Case for Geographical Push-Caching, in VINO: The 1994 Fall Harvest, *Technical Report TR-34-94, Center for Research in Computing Technology, Harvard University,* December, 1994.
19. Joseph D. Horton, and Alejandro López-Ortiz. On the Number of Distributed Measurement Points for Network Tomography, In Proceedings of *ACM Internet Measurements Conference* (IMC), pp. 204-209, 2003.
20. Sandy Irani. Page Replacement with Multi-Size Pages and Applications to Web Caching. In Proceeedings of *ACM Symposium on the Theory of Computing* (STOC) 1997, pp. 701–710.
21. S. Irani. Competitive Analysis of Paging. *Online Algorithms*, Lecture Notes in Computer Science 1441, Springer, 1998, pp.52-71.
22. S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang. On the Placement of Internet Instrumentation. *IEEE Conference on Computer Communications (INFOCOM)*, 2000, pp. 295-304.
23. S. Kalidindi and M. J. Zekauskas. Surveyor: An infrastructure for Internet performance measurements. *Proceedings of the 9th Internet Society Conference (INET),* ISOC, 1999.
24. Jussi Kangasharju, James W. Roberts, Keith W. Ross. Object replication strategies in content distribution networks. *Computer Communications*, 2002, vol. 25, no.4, pp. 376–383.
25. Magnus Karlsson and Mallik Mahalingam. Do We Need Replica Placement Algorithms in Content Delivery Networks. In Proceedings of *7th International Workshop on Web Content Caching and Distribution* (WCW), 2002.
26. Magnus Karlsson, Christos Karamanolis, and Mallik Mahalingam. *A unified framework for evaluating replica placement algorithms.* Technical Report HPL-2002, HP Laboratories, July 2002.
27. David R. Karger, Eric Lehman, Frank Thomson Leighton, Rina Panigrahy, Matthew S. Levine, Daniel Lewin. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. Proceedings of *ACM Symposium on the Theory of Computing* (STOC) 1997, pp.654–663.

28. P. Krishnan, Danny Raz, and Yuval Shavitt. The cache location problem. *IEEE Transactions on Networking*, vol. 8, no. 5, 2000, pp. 568–582.
29. Ritesh Kumar, and Jasleen Kaur. Efficient Beacon Placement for Network Tomography. In Proceedings of *ACM Internet Measurements Conference* (IMC), 2004.
30. Bo Li, Mordecai J. Golin, Giuseppe F. Italiano, Xin Deng and Kazem Sohraby. On the optimal placement of web proxies in the internet. Proceedings of *IEEE Conference on Computer Communications* (INFOCOM) 1999, pp.1282–1290.
31. A. López-Ortiz and D. M. Germán, A Multicollaborative Push-Caching HTTP Protocol for the WWW, *Poster proceedings of the 5th International World Wide Web Conference* (WWW96), 1996.
32. Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. Towards an accurate AS-level traceroute tool. In Proceedings of the *ACM Conference on Applications, technologies, architectures, and protocols for computer communications* (SIGCOMM), 2003.
33. Zhuoqing Morley Mao, David Johnson, Jennifer Rexford, Jia Wang, and Randy Katz. Scalable and Accurate Identification of AS-Level Forwarding Paths. Proceedings of *IEEE Conference on Computer Communications* (INFOCOM), 2004.
34. Evangelos P. Markatos. Main Memory Caching of Web Documents. Proceedings of the *Fifth International World Wide Web Conference* (WWW96), 1996.
35. V. Paxson. Measurements and Analysis of End-to-End Internet Dynamics. *PhD thesis, Univ. of Cal., Berkeley,* 1997.
36. V. Paxson, J. Mahdavi, A. Adams and M. Mathis, An Architecture for Large-Scale Internet Measurement. *IEEE Communications*, v.36, n.8, 1998, pp. 48–54.
37. Tim Roughgarden, and Éva Tardos. How Bad is Selfish Routing?. In Proceedings of *IEEE Symposium on Foundations of Computer Science* (FOCS) 2000, pp. 93–102.
38. J. Sedayao. Mosaic Will Kill My Network! Studying Network Traffic. Proceedings of the *First International Conference on the World-Wide Web* (WWW94), Elsevier, May, 1994.
39. R. Siamwalla, R. Sharma, and S. Keshav. Discovering Internet Topology. Technical Report, Cornell University, July 1998.
40. Cooperative Association for Internet Data Analysis (CAIDA). *The Skitter Project.* `http://www.caida.org/tools/measurement/skitter/index.html`, 2001.
41. D. Towsley. Network tomography through to end-to-end measurements. Abstract in *Proc. 3rd Workshop on Algorithm Engineering and Experiments (ALENEX),* 2001.
42. Berthold Vöcking. Selfish Routing and Congestion Games: Towards a game based analysis of the Internet. *School of Algorithmic Aspects of Game Theory in Large Scale Networks*, Santorini, Greece, 2004. `http://www.cti.gr/AAGTLSN/#LECTURERS`.