ACM SIGACT News Distributed Computing Column 18

Sergio Rajsbaum*

Abstract

The Distributed Computing Column covers the theory of systems that are composed of a number of interacting computing elements. These include problems of communication and networking, databases, distributed shared memory, multiprocessor architectures, operating systems, verification, Internet, and the Web. This issue consists of:

- "Algorithmic Foundations of the Internet," a survey by Alejandro López-Ortiz.

Many thanks to Alex for his contribution to this issue.

Request for Collaborations: Please send me any suggestions for material I should be including in this column, including news and communications, open problems, and authors willing to write a guest column or to review an event related to theory of distributed computing.

Algorithmic Foundations of the Internet

Alejandro López-Ortiz [1]

Abstract

In this paper we survey the field of Algorithmic Foundations of the Internet, which is a new area within theoretical computer science. We consider six sample topics that illustrate the techniques and challenges in this field.

1 Introduction

During the last 25 years the Internet has grown from being a small academic network connecting a few computer science departments to its present size, connecting more than 285 million computers and serving over 800 million users worldwide [Internet Software Consortium 2004].

With the introduction of the Mosaic web browser in 1993, the rate of growth accelerated both in terms of traffic and in number of users. It became clear then that the Internet would be large enough to make small performance differences significant. For example, for many realistic problem sizes outside the Internet, the difference between a $\Theta(n \log n)$ and a $\Theta(n)$ algorithm is not terribly significant, since $\log n$ is a such a slowly growing function. This is particularly so if the $\Theta(n \log n)$ solution has a simpler implementation resulting in a smaller leading constant. In contrast $\log_2 n$

*Instituto de Matemáticas, UNAM. Ciudad Universitaria, Mexico City, D.F. 04510 `rajsbaum@math.unam.mx`.

[1]School of Computer Science, University of Waterloo, Waterloo, ON, Canada. `alopez-o@uwaterloo.ca`

on the Internet is most often larger than 20 and in some instances as large as 50. In such cases, for example, a solution requiring n bits of storage is twenty or possibly fifty times smaller than a solution requiring space $n \log n$. This is by all means a non-negligible factor.

Algorithmic foundations of the Internet is a new field within theoretical computer science designed to address the challenges posed by such large scale networks. It lies in the intersection of networks, graph theory, algorithms and game theory. Other areas of relevance are distributed computing, statistical physics, economics, information retrieval and probability theory. Its early beginnings coincide with the popularization of the world wide web—and hence the Internet—during 1993-1995, and has been a rapidly expanding area of research ever since.

This field is, in a sense, a "theory of internetworking" [2]. This new theory of networking brings to bear tools that are of common use, so to speak, within the field of algorithms and theoretical computer science and applies them to networking problems. In particular it focuses on problems posed by the current Internet given its size and the specific set of protocols defining it. Early efforts in this field, as we shall see, were mostly algorithmic, e.g. how to compute better routing algorithms, shortest path distributed computations, better indexing algorithms for search engines, etc. Yet, as the field has grown it has incorporated techniques from graph theory, economics and game theory, statistics and probability, as well as statistical physics.

As well the field has received ample coverage from the popular press. This is in part due to the elegance of some of the results and in part to the fact that results can often be explained to the general public in approachable terms. For an enjoyable introduction some of the topics in this survey, see for example [Albert et al. 2000, Barabasi et al. 2000, Barabasi 2001, Barabasi and Bonabeau 2003, Chakrabarti 1999a, Chakrabarti 1999b, Jeong et al. 2000].

1.1 Historical Perspective

During the first fifteen years of the Internet, its initial design generally proved highly scalable and its growth was accommodated organically. Only in the early 1980's—a full decade after the network had first gone live in a form reasonably approaching the current Internet—did the first hints appear of the need to redesign certain aspects to accommodate impending growth. In particular the method for translating (or resolving) a computer name into an IP address started to show strains due to rapid network growth. Hence in 1984, the Domain Name System (DNS) was introduced. This system translates addresses such as `cs.uwaterloo.ca` to the numeric address `129.97.152.134`. When the DNS system was activated in 1984 it was one of earliest, largest and highest-transaction-rate distributed databases yet. It would take many years before any other in-production distributed database approached the size and transaction-rate of DNS.

Similarly, routing, which is the method used to direct a message across the network, showed strains in the late 1980's. Until then each computer in the network had something approaching a global map of the network in local storage. With the number of hosts rapidly growing such a method became unfeasible. Hence the Border Gateway Protocol (BGP) was introduced in 1989. The BGP protocol changed the routing decision process from a centralized global routing information model (EGP), to an innovative, decentralized protocol in which a node in the network is only responsible for knowing the local routing choice. This is equivalent to setting out to cross the entire continent, asking for directions at a service station and being told only what to do for the next five miles. The advantage is that each service station needs to know only their local neighbourhood; the disadvantage is that it leads to repeated queries throughout the route to the destination. BGP

[2] This should not be confused with the well established field of Queuing Theory which has traditionally been the source of theoretical foundations for communication networks.

introduced various innovations which were extremely well suited for theoretical analysis, and indeed some analysis took place early on, e.g. on the number of messages needed to disseminate routes starting from a blank network. Yet only a full five years after the BGP4 version of the protocol was introduced, did theoretical analysis show that, in the worst case, it could take as many as $\Theta(n!)$ steps to fully disseminate and stabilize a change on a network of n nodes [Labovitz at al. 2000].

The popularization of the World Wide Web in 1993 introduced yet another set of large scale challenges. For example, by 1995, the most advanced text indexing algorithms to that point had yet to meet their match within corporate applications. In contrast, by the end of 1995, those same algorithms were straining to sustain normal search engine loads on the Web. The situation has not improved much since, as all algorithmic advances in that field are quickly subsumed by the rapid growth in the web.

A second example is given by routing algorithms. The first routing algorithm devised had time complexity of $\Theta(\log n)$, which is generally considered fast as far as algorithms are concerned, yet today routers are strained using an improved $\Theta(\log \log n)$ algorithm. Again in most other settings the difference between those two time complexities would be too small to be of any impact. Yet on the Internet it is crucially significant.

1.2 Structure of the Survey

In this survey we have selected a subset of topics that are representative of the field of algorithmic foundations of the Internet. This selection of topics is by no means exhaustive, rather it was chosen as illustrative of the variety of algorithmic challenges arising from the Internet. Trimming down the set of choices was certainly a difficult task.

For each topic covered in the survey we start by describing at a high level what the problem or object of study is. While we assume certain familiarity with the general operation of the Internet, we describe each problem using layman's terms in as much as possible. We then give a brief historical perspective of how the challenge posed has been addressed so far within the network community, whenever relevant. This helps in introducing a more detailed view of what the specific problem is. Lastly we describe the theoretical and algorithmic aspects of the topic or challenge in question.

2 Web graph

The web graph is obtained by considering the web pages as nodes and the links between them as edges. That is, if a web page a links to a web page b, we introduce the directed edge (a, b) in the graph. Consider for a moment the various reasons why a link from a web page to another is created. It could be a recommendation, a related site, a reflection of structural information of the document (say a link from one chapter to the next in a book), a personal or professional relationship, etc. It follows then that the structure of the graph encodes valuable information for the study of semantic relationships, structure of information, topic classification, and information retrieval (see e.g. [Kleinberg et al. 1999, Bonato et al. 2005]).

The web graph has on the order of eight billion nodes and 60 billion edges and it is still rapidly growing. Given that the web is a publishing medium, it is likely that the graph will continue to grow and evolve. Furthermore, if we are to draw conclusions from the printing press, the pace of growth could remain as high as it is today for the foreseeable future.

The study of the web graph poses many problems. One of the better studied—albeit far from solved—is understanding the growth process of the graph. That is, what is the process by which nodes are created and how are links established? Once the growth process is properly modeled, it becomes possible to study which properties of the graph are expected to remain immanent over time and which are expected to change. E.g. should we expect the diameter of the web graph to be constant, increase or decrease over time?

A second important problem is how to sample the web graph. Naturally, if we had local access to the entire web graph, sampling it would be relatively straightforward. The problem is that web content is located in a distributed fashion across the world and there is no central directory of web pages. How do we discover then web pages on the Internet, and more importantly, how can we be assured that said pages are discovered in a representative fashion and not skewed in some unknown way. As examples of statistical properties of interest we have language distribution, and rate of update by percentiles i.e. pages that are updated daily, pages that are updated weekly, and so on. See [Henzinger 2004] for a more detailed discussion on this subject.

A third question is how to discover efficiently structures of interest in this enormous graph. For example, identifying bipartite subgraphs of the web graph is useful for web search engines. It turns out that for bipartite subgraphs one column corresponds to indexing pages that summarize sets of links on a given topic, while the other column corresponds to authorative sources that discuss said topic in depth (for a more detailed discussion see Section 5). Then, if a given query on a search engine matches all pages in the bipartite graph, it is natural to assume that a page from the authorative set is more liely to be a better answer than other pages which also contain the query terms but are not considered authorative. Given the size of the web graph, an $\Theta(n^2)$ algorithm for detecting such subgraphs is nearly impracticable, so this severely limits the universe of algorithmic solutions that can be considered. Indeed finding an efficient algorithm to identify all such bipartite graphs remains an open problem. A similar problem is to identify cliques in the web graph which represent communities of common interest. This problem in general is known to be NP-complete, hence any feasible algorithm would have to rely heavily on properties that are specific to the web graph. Lastly, as it is often the case, real life data is noisy, hence we need algorithms that are robust in the presence of noise. For example, assume we have a set of N pages for some large N, with all $N(N-1)/2$ possible edges edges between them except one. While this is not formally a clique, clearly it should be flagged as such for the purposes of a search engine.

3 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 [Sedayao 1994]. 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 significant impact on the bandwidth requirements of an organization.

The term caching denotes the management of 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 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

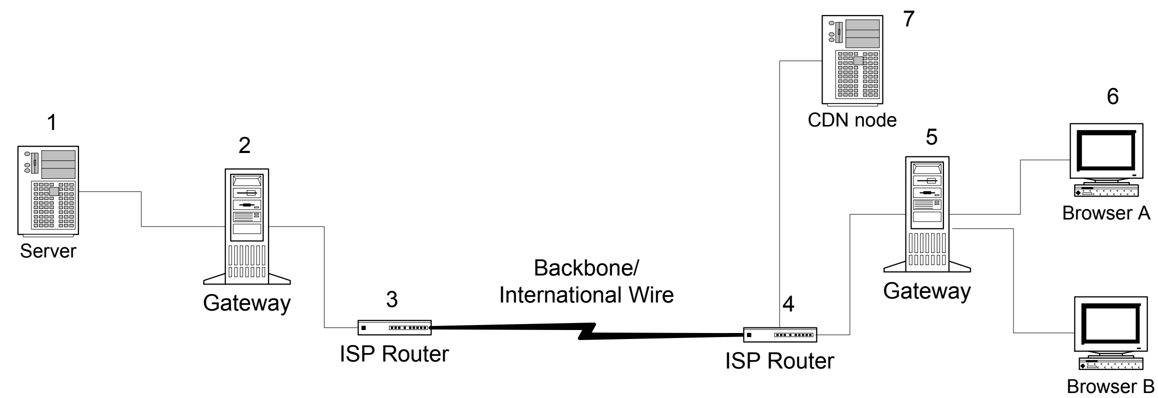

Figure 1: Web page access path over the network.

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.

3.1 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 organiza-

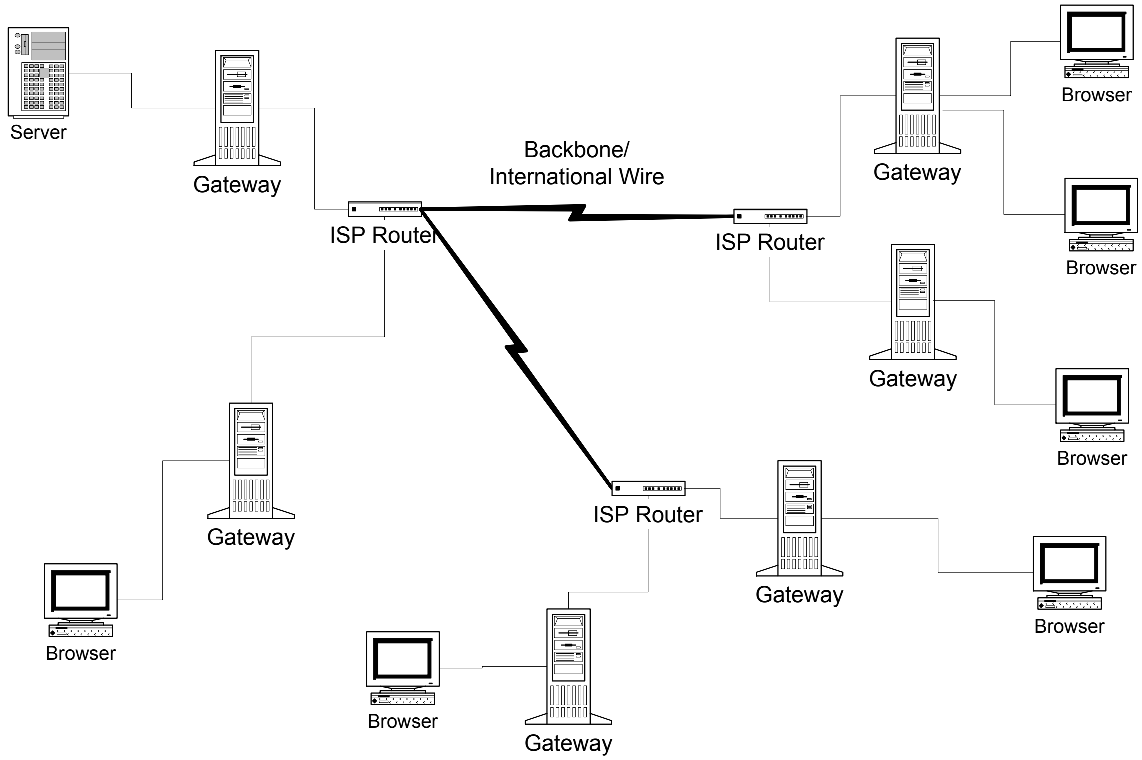

Figure 2: Network of distribution paths for a single server.

tional 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 [Gwertzman and Seltzer 1994], and subsequently supported by experimental setups [López-Ortiz and Germán 1996]. 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.

3.2 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 [Karger et al. 1997]. 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 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 [Krishnan et al. 2000] for arbitrary topologies of the graph, but it is known to be computable for tree topologies [Li et al. 1999]. 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 (e.g. [Kangasharju et al. 2002]).

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* [Kangasharju et al. 2002]. A general survey of the different algorithms can be found in [Karlsson et al. 2002] and an intriguing argument questioning the benefits of a coordinated policy over single server strategies appears in [Karlsson and Mahalingan 2002].

Web cache management strategies were first considered by [Markatos 1996] 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. 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 data 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 [Irani 1997, Cao and Irani 1997]. 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-

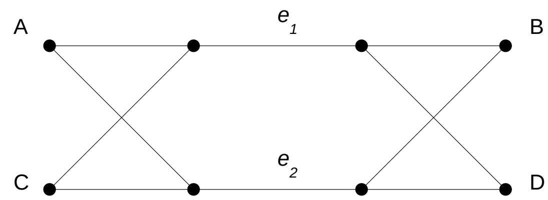

Figure 3: Simple network.

sized pages. Indeed, the offline optimum is NP-hard to compute under most models considered [Albers et al. 1999]. 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 [Irani 1997]. 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 [Cherkasova and Ciardo 2001, Golynski et al. 2003]. Further details on web caching can be found in [Irani 1998].

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), increased 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.

4 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.

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 they 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*.[3] 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[4].

Interestingly, it has been shown that the price of anarchy for the routing game is at most two in the worst case [Roughgarden and Tardos 2000]. 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 [Vöcking 2004] for a more detailed introduction to the subject of Internet routing and game theory.

5 Web information retrieval

The field of information retrieval has a long history going back to the early days of computer science. From its very beginnings there was an algorithmic component to this field, in the form of data structures and algorithms for text searching. For most of this time the state of the art in search algorithms stayed well ahead of the largest collection size. By the late 1980's with the computerization of the new edition of the Oxford English Dictionary the largest text collections could be searched in subsecond times. This situation prevailed until late 1995 when the web reached a size that would tax even the best indexing algorithms. Since then there has been a steady stream of theoretical and algorithmic challenges in web information retrieval.

5.1 Algorithmic Challenges in Web Information Retrieval

In terms of text indexing, to date the biggest challenge has been to device a search structure that supports searches for a keyword or pattern P in a text n bits long in time $|P|$ (i.e. proportional the

[3] 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.

[4] 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.

length of the pattern P) while using an index of size n (bit probe model). The state of the art in indexing techniques is very close to achieving this objective. The classic suffix tree structure supports searches in $|P|$ time but requires an index of size $O(n \log n)$. In contrast the recently developed compressed suffix arrays support searches in time $O(|P|/\log n + \log^{\epsilon} n)$ using $O(n)$ bits in the unit cost RAM model [Grossi and Vitter 1999, Manber and Myers 1993, Demaine and López-Ortiz 2003].

Most of the classic indexing schemes presume a model in which the user is searching for the specific location of a pattern in a contiguous text string. In contrast, search engine users give a collection of query terms (around three or so on average) and are searching for the subset of documents that contain some or all of the terms, ranked by some relevance metric. Observe that in this setting there is no inherent order between different documents, and the text position of the query terms is not required. In this model, the index is created at a document level granularity and the corpus is a collection of heterogeneous documents [Muthukrishnan 2002, Demaine et al. 2000].

A related problem consists of searching a large collection of documents that has been suitably tagged using XML. In such a setting the query language is extended to include predicates on the tags themselves, such as the XPath query language. For example [Geary et al. 2004] describes algorithms specifically tailored to XPath queries.

Aside from the algorithmic challenges in indexing and query processing, there are ranking and classification problems that are, to a certain extent, unique to web content. Web publishing is not centrally managed; as a result, content is of varied quality and duplicates of popular documents are common. At the same time, and in contrast to other heterogeneous collections, content is crossed-linked by means of links (`<A HREF="...">`). In practice it has been observed that this structure can be exploited to derive information on the relevance, quality, and even content, of a document. To give a trivial example, the number of incoming links to a page are a reflection of the popularity of the topic as well as the quality of the document. In other words, all other things being equal, the higher the quality and/or usefulness of the document, the more links it has. In practice this is not a very effective method to determine document relevance as popularity of the topic seems to occlude relevance. For example a very popular page that mentions an obscure term in passing will have more links that an authorative page defining and discussing the term.

5.2 Spectral Analysis

In 1998, Jon Kleinberg at IBM, and independently Larry Page et al. at Stanford, discovered a way to distinguish links due to popularity from those reflecting quality [Kleinberg 1998, Page et al. 1999]. The key observation was to consider first the subgraph of the web graph induced by the result set, and then consider the adjacency matrix A of that subgraph as a linear transformation. If one computes the eigenvalues of $A^T A$, they rank the pages by relative quality, as perceived by their peers. This presumes that when a web site links to another there is, to a large degree, an implicit endorsement of the quality of the content. If this is the case often enough, the eigenvalue computation will produce an efficient ranking of the web pages based purely on structural information.

It is difficult to overestimate the importance of this result. Traditionally, many, if not most, systems for ranking results were based on natural language processing. Unfortunately, natural language processing has turned out to be a very difficult problem, thus making such ranking algorithms impractical. In contrast, the eigenvalue methods used by Page's ranking and Kleinberg's HITS (hubs and authorities) do not require any understanding of the text and apply equally to pages written in any language, so long as they are HTML tagged and mutually hyperlinked.

A drawback of the hub and authorities method is that the amount of computation required at query time is impractically high. An open problem is to find an alternate method to compute the same or a similar ranking in a more efficient manner.

Interestingly the same eigenvalue computation can be used to categorize the pages by topic as well as perform other content-based analysis [Gibson et al. 1998, Azar et al. 2001]. This subfield has come to be known as spectral analysis of the web graph. [Azar et al. 2001] show that under a reasonable probabilistic model for the web graph, spectral analysis is robust under random noise scenarios.

Another interesting problem in information retrieval is the duplicate document/duplicate host problem. As we observed before, popular documents tend to be replicated numerous times over the network. If the document is in the public domain, users will make local copies available for their own benefit. Even if the popular content in question is proprietary, such as the web page of a news organization, it is likely to be duplicated by content distribution networks for caching purposes (see Section 3). In the case of public domain content, duplicate detection is complicated by the fact that users tend to change formatting and add extra headers and footers when creating a local copy. In the proprietary case, pages are usually duplicated verbatim and the challenge is to efficiently determine when the *entire* web site has been replicated and hence it does not need to be re-indexed. See [Henzinger 2004, Chakrabarti 1999b] which survey some of these problems in further detail.

6 Routing

The current Internet routing protocol (BGP4) uses a series of filters to determine the routing path for a given packet. That is, when a packet reaches a router with more than one outgoing link, there is a choice of which outgoing link is the correct one, if unique, or the preferred one if not unique. Such routing rules are given as a set of intervals in the IP address space.

To be more precise, recall that IP addresses are 4 octets long; hence they can be interpreted as a number between 1 and $2^{32}-1$. Since most organizations own a single consecutive range of IP addresses—or at most a few—it is sensible to declare rules that apply to a consecutive range of addresses or address interval.

BGP specifies that a routing rule or filter is given as a numeric interval of the form $[a \cdot 2^p, a \cdot 2^p + 2^k - 1]$, where a is binary number p digits long and $p + k = 32$. That is, all addresses sharing a as a prefix are routed in the same way. BGP allows for exceptions (holes) in a given such interval, with the exceptions themselves listed as a subinterval of the larger interval. The protocol allows for further exceptions to the exceptions, which are recursively expressed as subintervals of the subinterval. Interestingly, the exceptions do not need to be explicitly labeled as such. It suffices to specify that the routing rule to be followed for a given destination is that specified by the smallest interval containing the IP address of the destination. Note that intervals for filter rules are not allowed to overlap properly, i.e. if there is any overlap between two given intervals then one of the intervals must be fully contained in the other.

In practice a router in the core of the network would have a list of approximately 50,000 to 100,000 such intervals. A routing decision for each packet must be made within a few nanoseconds so as not to fall behind. Actual routers accomplish this by having a fast data structure to make the initial determination of which rule to apply, and thereafter by caching this result using special hardware (associative memory) that supports very fast accesses.

Under these constraints, it is of key importance that the data structure used for the initial determination to be particularly efficient. Earlier implementations used a variation of radix trees, which support lookups in time proportional to $\log n$, where n is the number of entries in the table. Since $n \approx 100,000$ we have that this method would require on the order of $\log_2 100,000 \approx$

17 operations which is too slow for our purposes. [Degermark et al. 1997] and simultaneously [Waldvogel et al. 1997] suggested the first data structures which support lookups in time $O(\log\log n)$.

These data structures, while described in quite different terms, are both essentially generalizations of the van Emde Boas data structure for lookups on universes of limited size. They are highly practical and indeed the methods of [Waldvogel et al. 1997] and its derivatives form the basis of the lookup data structures currently in use on the network. Surprisingly, most of the progress in this area has taken place within the context of applied network practitioners with relatively few contributions in theoreticians. [Muthukrishnan 2002] reviews some of these results and proposes a data structure which can filter packets in $O(1)$ time using $O(n^{1+\epsilon})$ space. The data structures above are presented in a static form, so a relevant question is how to devise a data structure that supports equally fast lookup times with relatively efficient update times [Ramesh et al. 2004].

Other related problems that have received some attention are conflict detection and table compression. Conflict detection is the process of verifying that filters are given as intervals which are properly nested (see e.g. [Muthukrishnan 2002, Baboescu and Varghese 2003]). Table compression is the simplification of the table resulting from identifying when an exception is redundant (i.e. both the original rule and the exception are the same) or if two consecutive exception intervals describe the same exception and hence can be merged into a single rule [Suri et al. 2003]. In practice, both of these operations are undertaken whenever a new routing table is entered into router memory by a network administrator.

7 Tomography

Efficient routing and caching require accurate connectivity information of the Internet. Internet protocols, on the other hand, make this task difficult as routing 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* [Claffy et al. 1998, Claffy et al. 1999, Towsley 2001].

7.1 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 [Paxson 1997]. 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) [Adams et al. 1998, Paxson et al. 1998, Francis et al. 1999] 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 [Caceres et al. 1999]. Other efforts along these lines are [Kalindi and Zekauskas 1999, Skitter 2001, Towsley 2001, Siamwalla et al. 1998, Francis et al. 1999]. The location of these beacons is determined according to various heuristics [Adams et al. 2000, Adler et al. 2001, Branigan et al. 2001, Cheswick et al. 2000].

7.2 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. 2002] showed that in the case of the high level AS topology, some of the techniques in use provided biased samples.

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 [Faloutsos et al. 1999]. 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 [Chen et al. 2002, Mao et al. 2003, Mao et al. 2004]. 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 level, as we shall see.

[Jamin et al. 2000] 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. Recently, [Barford et al. 2001] 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. 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. 2002] consider the problem of the effectiveness of tomography on networks of general topology. 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 $\Omega(\log n)$ [Horton and López-Ortiz 2003]. 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 and Kaur 2004] propose a refinement of the model which produces a robust beacon set of even smaller size.

8 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

[Adams et al. 2000] 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.

[Adams et al. 1998] 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.

[Adler et al. 2001] 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.

[Albers et al. 1999] 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.

[Albert et al. 2000] R. Albert, H. Jeong, and A.-L. Barabśi. Error and attack tolerance of complex networks. Nature 406, 378-482 (2000).

[Azar et al. 2001] Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry, and Jared Saia. Spectral Analysis of Data. In Proceedings of *ACM Symposium on Theory of Computing* (STOC), 2001, pp. 619–626.

[Baboescu and Varghese 2003] Florin Baboescu, and George Varghese. Fast and scalable conflict detection for packet classifiers. *Computer Networks* vol.42, no. 6, pp. 717–735, 2003.

[Barabasi 2001] A.-L. Barabási. The physics of the Web. *Physics World*, 14, 33 (2001).

[Barabasi et al. 2000] A.-L. Barabási, R. Albert, H. Jeong, and G. Bianconi. Power-Law Distribution of the World Wide Web, Science 287, 2115 (2000).

[Barabasi and Bonabeau 2003] A.-L. Barabási, and E. Bonabeau. Scale-free networks. *Scientific American*, 288(5):60, 2003.

[Barford et al. 2001] 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.

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

[Bu et al. 2002] 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

[Bestavros 1995] A. Bestavros, Demand-based Document Dissemination for the World-Wide Web. *Technical Report 95-003, Computer Science Department, Boston University.* February, 1995.

[Bestavros et al. 1995] A. Bestavros, R. L. Carter, M. E. Crovella, C. R. Cunha, A. Heddaya, and S. A. Mirdad. Application-Level Document Caching in the Internet. *Technical Report BU-CS-95-002, Computer Science Deparment, Boston University*, Revised March, 1995.

[Bonato et al. 2005] Anthony Bonato. A survey of models of the Web graph. To appear, in proceedings of the *Workshop on Combinatorial and Algorithmic Aspects of Networking 2004* (CAAN), Lecture Notes in Computer Science 3405, Springer Verlag, 2005.

[Caceres et al. 1999] 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.

[Cao and Irani 1997] Pei Cao, and Sandy Irani. Cost-Aware WWW Proxy Caching Algorithms. In Proceedings of *USENIX Symposium on Internet Technologies and Systems*, 1997.

[Chakrabarti 1999a] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Hypersearching the Web. *Scientific American*, June 1999.

[Chakrabarti 1999b] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Mining the link structure of the World Wide Web. *IEEE Computer*, August 1999.

[Chen et al. 2002] 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.

[Cherkasova and Ciardo 2001] 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.

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

[Claffy et al. 1998] 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.

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

[Dahlin et al. 1994] M. D. Dahlin, R. Y. Wang, T. E. Anderson, D.A. Patterson, Cooperative Caching: Using Remote Client Memory to Improve File System Performance. Proceedings of the *First Symposium on Operating Systems Design and Implementation (OSDI)*, pp.267-280, 1994.

[Demaine and López-Ortiz 2003] Erik D. Demaine, and Alejandro López-Ortiz. A linear lower bound on index size for text retrieval. *Journal of Algorithms*, Vol. 48, no. 1, pp. 2-15, 2003.

[Degermark et al. 1997] Mikael Degermark, Andrej Brodnik, Svante Carlsson, and Stephen Pink. Small Forwarding Tables for Fast Routing Lookups. In Proceedings of the *ACM Conference on Applications, technologies, architectures, and protocols for computer communications* (SIGCOMM), 1997. pp. 3–14.

[Demaine et al. 2000] Erik D. Demaine, Alejandro López-Ortiz, and J. Ian Munro. Adaptive set intersections, unions, and differences. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms* (SODA) 2000, pp. 743-752.

[Eppstein and Muthukrishnan 2001] David Eppstein, and S. Muthukrishnan. Internet packet filter management and rectangle geometry. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms* (SODA) 2001, pp. 827-835.

[Faloutsos et al. 1999] 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.

[Francis et al. 1999] 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

[Geary et al. 2004] Richard F. Geary, Rajeev Raman, and Venkatesh Raman. Succinct ordinal trees with level-ancestor queries. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms*, (SODA) 2004, pp. 1–10.

[Gibson et al. 1998] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In Proceedings of the *ACM Conference on Hypertext and Hypermedia*, 1998, pp. 225–234.

[Glassman 1994] S. Glassman, A Caching Relay for the World Wide Web, Proceedings of the *First International Conference on the World-Wide Web* (WWW94), Elsevier, May, 1994.

[Golding and Long 1991] R. Golding, and D. D. E. Long, Accessing Replicated Data in a Large-Scale Distributed System. *Technical Report 91-01, Concurrent Systems Laboratory, University of California, Santa Cruz*, January, 1991.

[Golynski et al. 2003] 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.

[Grossi and Vitter 1999] Roberto Grossi, and Jeffrey Scott Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching. In Proceedings of *ACM Symposium on Theory of Computing* (STOC), 1999, pp. 397–406.

[Gwertzman and Seltzer 1994] 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.

[Gwertzman 1995] J. Gwertzman. Autonomous Replication in Wide-Area Internetworks.*Technical Report TR-19-95, Center for Research in Computing Technology, Harvard University,* April, 1995.

[Henzinger 2004] Monika R. Henzinger. Algorithmic Challenges in Web Search Engines. *Internet Mathematics*, vol. 1, no. 1, 2004, pp. 115-126.

[Horton and López-Ortiz 2003] 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.

[Internet Software Consortium 2004] Internet Software Consortium `http://www.isc.org`.

[Irani 1997] 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.

[Irani 1998] S. Irani. Competitive Analysis of Paging. *Online Algorithms*, Lecture Notes in Computer Science 1441, Springer, 1998, pp.52-71.

[Jamin et al. 2000] 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.

[Jeong et al. 2000] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A.-L. Barabsi. The large-scale organization of metabolic networks. *Nature* 407, pp. 651–654, 2000.

[Jones 1994] R. Jones. Digital's World-Wide Web Server. A Case Study. Proceedings of the *First International Conference on the World-Wide Web* (WWW94), Elsevier, May, 1994.

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

[Kangasharju et al. 2002] 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.

[Karlsson and Mahalingan 2002] 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.

[Karlsson et al. 2002] Magnus Karlsson, Christos Karamanolis, and Mallik Mahalingam. *A unified framework for evaluating replica placement algorithms.* Technical Report HPL-2002, HP Laboratories, July 2002.

[Karger et al. 1997] 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.

[Kleinberg 1998] J. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms* (SODA), 1998, pp. 668–677.

[Kleinberg et al. 1999] J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a graph: Measurements, models and methods. *Invited survey at the International Conference on Combinatorics and Computing*, 1999.

[Krishnan et al. 2000] P. Krishnan, Danny Raz, and Yuval Shavitt. The cache location problem. *IEEE Transactions on Networking*, vol. 8, no. 5, 2000, pp. 568–582.

[Kumar and Kaur 2004] Ritesh Kumar, and Jasleen Kaur. Efficient Beacon Placement for Network Tomography. In Proceedings of *ACM Internet Measurements Conference* (IMC), 2004.

[Labovitz at al. 2000] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. In Proceedings of *ACM Conference on Applications, Technologies Architectures and Protocols for Computer Communication* (SIGCOMM) 2000, pp. 175–187.

[Li et al. 1999] 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.

[López-Ortiz and Germán 1996] 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.

[Manber and Myers 1993] U. Manber, and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, vol. 22, no. 5, 1993, pp. 935–948.

[Mao et al. 2003] 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.

[Mao et al. 2004] 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.

[Markatos 1996] Evangelos P. Markatos. Main Memory Caching of Web Documents. Proceedings of the *Fifth International World Wide Web Conference* (WWW96), 1996.

[Muthukrishnan 2002] S. Muthukrishnan. Efficient algorithms for document retrieval problems. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms*, (SODA) 2002.

[Page et al. 1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Technical Report, Department of Computer Science, Stanford University*, 1999-66.

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

[Paxson et al. 1998] 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.

[Pitkow and Recker 1994] J. E. Pitkow, M. M. Recker, A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns. Proceedings of the *First International Conference on the World-Wide Web* (WWW94), Elsevier, May, 1994.

[Ramesh et al. 2004] Priyank Ramesh Warkhede, Subhash Suri, and George Varghese. Multiway range trees: scalable IP lookup with fast updates. *Computer Networks*, vol. 44, no. 3, pp. 289–303, 2004.

[Roughgarden and Tardos 2000] 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.

[Sedayao 1994] 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.

[Siamwalla et al. 1998] R. Siamwalla, R. Sharma, and S. Keshav. Discovering Internet Topology. Technical Report, Cornell University, July 1998.

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

[Suri et al. 2003] Subhash Suri, Tuomas Sandholm, Priyank Ramesh Warkhede. Compressing Two-Dimensional Routing Tables. *Algorithmica*, vol. 35, no. 4, pp. 287–300, 2003.

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

[Vöcking 2004] 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.`

[Waldvogel et al. 1997] Marcel Waldvogel, George Varghese, Jon Turner, and Bernhard Plattner. Scalable High Speed IP Routing Lookups. In Proceedings of the *ACM Conference on Applications, technologies, architectures, and protocols for computer communications* (SIGCOMM), 1997. pp. 25–36.