My Ten Favorite “Practical Theory” Papers

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

As the saying goes, “In theory there is no difference between theory and practice. But, in practice, there is.” Networking research has a wealth of good papers on both sides of the theory-practice divide. However, many practical papers stop short of having a sharp problem formulation or a rigorously considered solution, and many theory papers overlook or assume away some key aspect of the system they intend to model. Still, every so often, a paper comes along that nails a practical question with just the right bit of theory. When that happens, it’s a thing of beauty. These are my ten favorite examples. In some cases, I mention survey papers that cover an entire body of work, or a journal paper that presents a more mature overview of one or more conference papers, rather than single out an individual research result. (As an aside, I think good survey papers are a wonderful contribution to the community, and wish more people invested the considerable time and energy required to write them.)

Categories and Subject Descriptors

C.2.2 [**Internetworking**]: Network protocols

General Terms

Algorithms, measurement, performance, theory

Keywords

Protocols, routing, optimization, measurement, scheduling, load balancing

1. PROTOCOL DESIGN AND ANALYSIS

These two papers illustrate beautifully that a protocol is (or should be!) a distributed solution to some well-formulated problem. Often, the theoretical model comes after the fact, as a faithful act of reverse engineering. Perhaps, in the future, we can “play it forward” and use models to guide how we design protocols in the first place.

1. T. G. Griffin, F. B. Shepherd, and G. Wilfong, “The stable paths problem and interdomain routing,” *IEEE/ACM Transactions on Networking*, pp. 232–243, April 2002.

This journal paper, and the conference papers leading up to it, got me (and many other people) interested in BGP. The paper describes the problem that BGP is (implicitly) solving and proved a number of disturbing negative results about the interdomain routing system. The paper chipped away at the considerable minutia in BGP to distill the most important aspects of the protocol—that each node has its own local ranking of (some subset of) the paths, and picks the highest ranked path consistent with its neighbors’ choices. The paper also presents numerous simple examples—with clever names like Bad Gadget, Surprise, and Disagree—to illustrate key ideas and counterexamples. And, who would have ever thought there was a connection between BGP and 3-SAT? The paper has fostered a wealth of theoretical work on BGP in recent years—research that would likely have never happened without someone first doing the heavy lifting to create a crisp, accurate model of the protocol. This paper is by no means an easy read, but it is well worth the effort.

2. M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proceedings of the IEEE*, pp. 255–312, January 2007.

This paper surveys the growing body of work on designing and analyzing protocols as distributed solutions to optimization problems. The paper puts mathematical rigor around the elusive notion of “network architecture”—the definition and placement of function in a network. Decomposing the optimization problem leads to a collection of subproblems (corresponding to different protocol layers) and variables coordinating the subproblems (corresponding to the interfaces between layers). For example, early work showed that TCP congestion control—where end hosts increase and decrease their sending rates in response to packet losses—implicitly maximizes aggregate user utility. Each of the many variants of TCP correspond to a utility function with a different shape. Using optimization decomposition for “forward engineering” has led to a variety of new protocols, including TCP FAST and new MAC protocols, with provable optimality and stability properties. Although optimization theory does not provide all the answers, this paper shows that it can be a valuable guide along the way.

2. NETWORK-WIDE MEASUREMENT

During the last ten years, network measurement has become a rich and vibrant research area, starting with early work characterizing the properties of IP traffic on individual links and the performance of end-to-end paths through the Internet. Later work recognized the value of measurement data in the design and operation of data networks, such as Internet Service Provider backbones. Network operators often need a *network-wide* view of the traffic, to detect denial-of-service attacks or perform traffic engineering. However, there is often a large gap between the available measurement data and what the network operators want to know. These two papers offer a fresh view of how to extract the most useful information out of a small amount of measurement data.

3. N. G. Duffield and M. Grossglauser, “Trajectory sampling for direct traffic observation,” *IEEE/ACM Transactions on Networking*, pp. 280–292, June 2001.

Most measurement research has focused on making clever use of whatever data we are lucky enough to have, or designing point solutions that compute one specific statistic of interest. This paper is a notable exception. The paper proposes that routers sample packets in a consistent fashion, to compute a "trajectory" that follows a subset of the packets as they flow through the network. Trajectory sampling can be used to determine the application mix in a network, trace distributed denial-of-service attacks, measure one-way loss and delay, and so on. Rather than marking the sampled packets as they enter the network, each router applies the same hash function to sample the same packets at each hop in their journey. The paper shows that IP packets have sufficient entropy to enable pseudo-random sampling with a hash function that operates on a relatively small number of bits. A second hash function can produce concise summaries of the sampled packets for efficient export to the monitoring system. Trajectory sampling is simple and effective, and is part of the packet sampling standards being developed in the "psamp" working group at the IETF.

4. Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *Proc. ACM SIGMETRICS*, pp. 206–217, June 2003.

Network operators need to know the *traffic matrix*—the offered load between all pairs of ingress and egress points in the network—to perform traffic engineering and capacity planning. However, the traffic matrix is surprisingly difficult to compute without fine-grained measurements around the perimeter of the network. For several years, researchers investigated "tomography" techniques that try to infer the traffic matrix from link-load statistics and routing information. However, the problem is significantly under-constrained, since the number of links (providing aggregate load statistics) is much less than the number of ingress-egress pairs (comprising the traffic matrix). Early research tried to apply existing tomography techniques (developed in other domains, such as transportation networks), but the results were fairly poor due to a mismatch in the assumptions underlying these models. This paper introduced a "gravity model" that worked much better, and faster, than previous techniques. An excellent follow-up paper on "An information-theoretic approach to traffic matrix estimation" at *SIGCOMM'03* provided the mathematical explanation for why the so-called "tomo-gravity" scheme works so well.

3. EFFICIENT DATA STRUCTURES

Many networking problems are, at heart, problems of scale. Often, we have more data than we can efficiently analyze, or need to maintain more state than is reasonable to store in practice. Clever ways to cull through large volumes of data, or maintain an accurate approximation of system state, are hugely valuable in practice. These two papers are nice examples.

5. Cristian Estan, Stefan Savage, and George Varghese, "Automatically inferring patterns of resource consumption in network traffic," *Proc. ACM SIGCOMM*, pp. 137–148, August 2003.

Network operators analyzing packet or flow traces must extract meaningful information from large volumes of high-dimensional data. Early work simply aggregated the data in pre-determined ways (such as TCP connections, IP address pairs, or IP prefixes) and reported the top contributors of traffic. However, the network operators actually want something different: they want to know the "important" or "unexpected" contributors of traffic. This paper presents algorithms that identify large traffic clusters, with the goal of minimizing their representation. The algorithms compute multi-dimensional trees that aggregate the data along various dimensions, and identify the nodes that represent significant and distinctive aggregates of traffic. For example, the algorithms might identify that a single UDP transfer is responsible for 15% of the traffic, and a particular source IP address is responsible for another 25%, rather than simply reporting a list of the top-ten flows. The paper presents algorithms and bounds, evaluation on measurement traces, and a publicly-released tool (called AutoFocus) used by network operators.

6. A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Mathematics*, vol. 1. no. 4, pp. 485–509, 2004.

A Bloom filter is a succinct data structure for representing a set of items, with the goal of answering membership queries. A Bloom filter often requires significantly less space and bandwidth than storing and transmitting an entire list of items, at the cost of introducing false positives. (Fortunately, false positives are acceptable in many practical applications.) The basic idea is to maintain an array of m bits and use k independent hash functions to map each item to a random number in the range $1, 2, \ldots, m$. Adding an element to the set involves setting the associated bits in the Bloom filter to 1; similarly, a set-membership query involves checking that all of the associated bits are set to 1. Although Bloom filters were invented more than 35 years ago, their use in the networking community is relatively new. This survey paper gives a nice overview of the ways Bloom filters have been used, and extended, in solving a variety of networking problems, such as summarizing content in peer-to-peer networks, collecting traffic measurements, and detecting forwarding loops. The paper also gives a very accessible overview of the mathematics behind Bloom filters and several useful variants like counting Bloom filters and compressed Bloom filters.

4. TRAFFIC SHAPING AND SCHEDULING

In the early-to-mid 1990s, the networking community produced a wealth of interesting theoretical papers on providing quality-of-service guarantees in packet-switched networks. Narrowing down to a small set of papers is understandably difficult, but these two papers stick out in my mind as great examples of clearly formulated practical problems coupled with elegant, rigorous solutions.

7. A. Parekh and R. Gallagher, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking*, pp. 344-357, June 1993.

This paper and the sequel on the "multiple-node case" (in the April 1994 issue of *ToN*) are true classics. I've lost count of how many times I read them back in the mid 1990s. The paper proposes generalized processor sharing (GPS) as an idealized model for the design and analysis of packet-scheduling algorithms such as weighted fair queuing (WFQ). In GPS, each active flow i is allocated a share of the link bandwidth in proportion to its weight w_i, where all traffic is fluid. Practical WFQ schemes can schedule packets based on their finishing time (or starting time) in a GPS simulation of the system, and their performance can be analyzed in terms of the fairness and delay guarantees the idealized GPS system would have provided. In particular, the paper shows that packet-based GPS schemes, combined with leaky-bucket admission control, allow a network to provide tight bounds on throughput and delay. Neat.

8. J. D. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Transactions on Networking*, pp. 397–410, August 1998.

Variable-bit-rate video streams, with frame sizes that can vary wildly over time, are challenging to deliver efficiently. Allocating network bandwidth for the peak bit rate is wasteful, but allocating for the mean is not sufficient. During the mid 1990s, several papers explored ways to "smooth" the variable-bit-rate video by capitalizing on buffer space at the receiver. The basic problem is to transmit frame i (with size f_i) to a receiver with buffer space b in time for playback, while minimizing the burstiness of the network traffic. The sender must compute a transmission schedule that delivers enough data to support continuous playback, but not so much that the receiver's buffer overflows. The paper proposed a linear-time algorithm for computing the schedule that minimizes the maximum transmission rate and all higher-order moments, leading to the greatest possible reduction in rate variability. In addition to presenting an optimal solution to a practical problem, the paper made an interesting connection to the theory of *majorization* to formally define "smoothness" and prove that the algorithm is optimal.

5. LOAD BALANCING

These last two papers are not really networking papers at all, but I feel compelled to include them in my top-ten list because the lessons they teach are so relevant to networking. Load balancing is a problem that arises in many areas of computer science, ranging from managing jobs on a shared computer cluster to splitting data packets over multiple paths through a network. Load balancing raises important questions about how much flexibility is necessary to make efficient use of system resources, whether it is worthwhile to revisit past load-balancing decisions, and how to prevent oscillation when reacting to stale information. These two papers focus precisely on these questions.

9. Mor Harchol-Balter and Allen Downey, "Exploiting process lifetime distributions for dynamic load balancing," *ACM Transactions on Computer Systems*, pp. 253–285, August 1997.

This paper challenged conventional wisdom on whether it is worthwhile to migrate a running job to a different machine to balance load in a computer cluster. Earlier work had argued that the load-balancing benefits are outweighed by the overhead of moving the job, suggesting that load balancing should focus solely on placing *new* jobs as they enter the system. The paper analyzed traces of UNIX workstation workloads to show that, in practice, job lifetimes have very high variance, with a few jobs running for quite a long time. Once a job has stayed in the system for a while, it is very likely to stay for a long time. Migrating these long-running jobs is well worth the effort, as the analytical models in the paper demonstrated. The paper showed, quite effectively, that the workload can have a profound influence on the appropriate design of a system. For me, the paper had another valuable lesson—that high variability in the workload is not necessarily a bad thing, and can in fact be turned to your advantage.

10. M. Mitzenmacher, A. Richa, and R. Sitaraman, "The power of two random choices: A survey of techniques and results," Book chapter, in *Handbook of Randomized Computing: Volume 1*, edited by P. Pardalos, S. Rajasekaran, and J. Rolim, pp. 255–312, 2001.

To be honest, I included this survey paper in my top-ten list as a somewhat sneaky way to mention *eleven* papers, as I wanted to mention two papers that I like a lot: "The power of two choices in randomized load balancing" (*IEEE Transactions on Parallel and Distributed Computing*, October 2001) and "How useful is old information?" (*IEEE Transactions on Parallel and Distributed Systems*, January 2000), both by Michael Mitzenmacher. The survey paper summarizes both of these papers, and more. The first paper, on the power of two choices, shows that even a little flexibility goes a long way. In particular, choosing the best of two randomly-selected choices (say, of paths through a network) can be remarkably effective. The second paper, on stale information, explores how to balance load when the information is out of date. Selecting the "best" choice can be misguided, and even lead to instability. Instead, selecting the best of two randomly-chosen options is a better approach. Over the years, I have found both of these papers very helpful in thinking about how to make load-sensitive routing stable and efficient.