Randomized Network Algorithms: An Overview and Recent Results

Balaji Prabhakar
Stanford University

I. Extended Abstract

While the design and analysis of algorithms is a classical subject, networking applications give rise to some unique constraints that often render well-known optimal solutions either prohibitively expensive or impossible to implement. Essentially, in many situations, optimal solutions suffer from the “curse of dimensionality:” they do not scale well to high speeds and/or to large user populations. In such cases one needs to invent approximate schemes that work within the available technology and yet perform very well. Randomization is one method for building approximate algorithms that has proved very successful. We overview the basic idea of randomization and some recent work.

The main idea of randomized algorithms is simple to state: Basing decisions upon a few randomly chosen samples is a good surrogate for basing decisions upon the complete state. Therefore, randomized algorithms lead to the simple implementation of otherwise complicated solutions. To understand this principle in a concrete setting, consider randomized load balancing. Jobs arrive at a bank of n independent, rate-1 exponential servers according to a Poisson process of rate $n\lambda$, $\lambda < 1$. The problem is to assign the arriving jobs to one of the queues so that backlogs are kept small. The Join the Shortest Queue (JSQ) policy, which assigns an arriving job to the shortest queue (breaking ties at random), gives very small backlogs. However, it requires the determination of the shortest queue which can be complex. Vvedenskaya et. al. [1] and Mitzenmacher [2] analyzed randomized policies where the server allocates an arriving job to the shortest of $d \geq 1$ randomly chosen queues. They showed that when $d \geq 2$ the tail of the backlog process is *superexponentially distributed* as compared to just exponential tails for $d = 1$. Thus, randomized algorithms can be quite powerful in their simplicity and performance.

We consider randomized algorithms in various other contexts: switch scheduling, web caching and bandwidth partitioning. Since the performance of a randomized algorithm depends crucially on the quality of the samples, we ask: (a) Is it possible to improve the quality of the samples *without* increasing their *number*? (b) If yes, how well would such an improvement perform? Considering these questions within the context of algorithms for web caching [3], switch scheduling [4] and load balancing [5], we describe and analyze the performance of a simple trick for recursively improving the sample quality, whilst leaving its size fixed. This trick yields a significant performance boost while retaining the essential simplicity of randomized schemes.

Finally, we briefly describe a randomized bandwidth partitioning algorithm, called CHOKe (Choose and Keep or Choose and Kill), for approximately fairly partitioning the link bandwidth at a congested router among the flows traversing the link [6]. A further refinement of this scheme, called AFD (Approximate Fair Drop), has been implemented in several commercial switch and router platforms.

References

[1] N.D. Vvedenskaya, R.L.Dobrushin and F.I.Karpelevich: “Queueing system with selection of the shortest of two queues : An asymptotic approach,” *Problems of Information Transmission*, **32**:15–27, 1996.

[2] Mitzenmacher: “The power of two choices in randomized load balancing,” *PhD thesis*, University of California, Berkeley, 1996.

[3] K. Psounis, B. Prabhakar: “Efficient Randomized Web-Cache Replacement Schemes Using Samples from Past Eviction Times,” *IEEE/ACM Trans. on Networking*, 10(4):441-454, August 2002.

[4] P. Giaccone, B. Prabhakar, D. Shah: “Randomized scheduling algorithms for input-queued switches,” *IEEE Journal on Selected Areas in Communication*, 21(4), 2003.

[5] D. Shah, B. Prabhakar: “The use of memory in randomized load balancing,” *Int. Symp. on Information Theory, 2002.*

[6] R. Pan, B. Prabhakar, K. Psounis: “CHOKe - A stateless active queue management scheme for approximating fair bandwidth allocation,” *Proc. of Infocom 2000.*