

Philip Brighten Godfrey

RAD Lab
465 Soda Hall
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720-1776

pbg@cs.berkeley.edu
<http://www.cs.berkeley.edu/~pbg/>
Phone: 650-814-1962
Fax: 510-642-5775

Research Interests

Networking and distributed systems: in particular, routing architecture, algorithms for and analysis of distributed systems, and applications of theory to systems design.

Education

- | | |
|----------------------|--|
| Expected
May 2008 | University of California at Berkeley , Ph.D. in Computer Science
Dissertation title: "Designing Distributed Systems for Heterogeneity"
Advisor: Prof. Ion Stoica
Major in CS Systems, minors in CS Theory and Statistics |
| December 2006 | University of California at Berkeley , M.S. in Computer Science
Thesis title: "Minimizing Churn in Distributed Systems"
Advisor: Prof. Ion Stoica |
| May 2002 | Carnegie Mellon University , B.S. in Computer Science
Minors in Jazz Performance and Trumpet Performance |

Awards and Honors

- Authored grant proposal with Ion Stoica (PI), "Stabilizing BGP, Safely", awarded support of \$98,700 by the Cisco Collaborative Research Initiative (CCRI) in January 2008.
- NSF Graduate Research Fellowship, 2004-2007.
- California Microelectronics Fellowship, 2002-2003.
- Phi Kappa Phi honor society, Spring 2002.
- Honorable Mention, Computing Research Association Outstanding Undergraduate Award Program, 2002. One of 44 from the US and Canada chosen for "outstanding research potential in an area of computing research."
- Honorable Mention, 2002 Google Scholarship. One of eight US students recognized.
- 2002 Andrew Carnegie Society Presidential Scholar. One of 34 recognized students in my graduating class.
- Phi Beta Kappa, October 2001 (early induction). One of 20 early inductees in my graduating class.
- CMU Small Undergraduate Research Grant, Spring 2002. Awarded by CMU's Undergraduate Research Initiative, and sponsored by Compaq Computer Corporation, for my research in Natural Language CAPTCHAs (see below).

Employment

- December 2006-March 2007 **Consultant, Rinera Networks, Inc.**
Design and simulation of algorithms for a component of a peer-to-peer video distribution system.
- August 2002-present **Graduate Student Researcher, University of California - Berkeley**
Advised by Prof. Ion Stoica, researched how distributed systems are affected by and can take advantage of *heterogeneous reliabilities* among the system's components, including development of methods to safely avoid failures in Internet routing, and a study of principles of minimizing churn applicable to many distributed systems which showed how randomization is an effective technique for avoiding failures. Modeled the effect of *capacity heterogeneity* on parallel systems, and designed a distributed hash table which effectively and efficiently adapts to heterogeneous nodes.
- May-August 2004 **Research intern, Intel Research Pittsburgh**
As part of the Open DHT project, designed and implemented a new version of the ReDiR rendezvous algorithm which automatically adapted to group size. Supervisor: Brad Karp.
- May-August 2002 **Engineering intern, Google Inc.**
Clustering of data for Froogle, a product search tool.
- August 2001 - May 2002 **Undergraduate researcher, CAPTCHA project, Carnegie Mellon University**
Research in methods to automatically differentiate humans and computers using a natural language-based (i.e., text only) test. Advisor: Prof. Lenore Blum. See <http://captcha.net>.
- May-August 2001 **Engineering intern, Cray Inc.**
Design and implementation in C of a fast multi-link file transfer protocol and server for the Cray X1 supercomputer.
- May-August 2000 **Engineering intern, Cray Inc.**
Development of routing software for the Cray X1 memory subsystem. Involved translating a prototype from LISP to C++ and work on the routing algorithm.

Publications

Available at <http://www.cs.berkeley.edu/~pbg/>.

- P. Brighten Godfrey and Richard M. Karp. On the Price of Heterogeneity in Parallel Systems. To appear, *Theory of Computing Systems*. DOI: 10.1007/s00224-008-9102-5. Invited journal version of earlier SPAA 2006 publication.
- P. Brighten Godfrey. Balls and Bins with Structure: Balanced Allocations on Hypergraphs. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.
- P. Brighten Godfrey, Matthew Caesar, Ian Haken, Scott Shenker, and Ion Stoica. Stable Internet Route Selection. *Meeting of the North American Network Operators' Group (NANOG 40)*, June 2007.
- Alexandros G. Dimakis, P. Brighten Godfrey, Martin J. Wainwright, and Kannan Ramchandran. Network Coding for Distributed Storage Systems. *IEEE INFOCOM*, May 2007.

- Alexandros G. Dimakis, P. Brighten Godfrey, Martin J. Wainwright, and Kannan Ramchandran. The Benefits of Network Coding for Peer-to-Peer Storage Systems. *Third Workshop on Network Coding, Theory, and Applications (NETCOD)*, January 2007.
- P. Brighten Godfrey, Scott Shenker, and Ion Stoica. Minimizing Churn in Distributed Systems. *ACM SIGCOMM*, September 2006.
- P. Brighten Godfrey and Richard M. Karp. On the Price of Heterogeneity in Parallel Systems. *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, July 2006.
- Sonesh Surana, Brighten Godfrey, Karthik Lakshminarayanan, Richard Karp and Ion Stoica. Load Balancing in Dynamic Structured P2P Systems. *Performance Evaluation*, Vo. 63, no. 6, pps. 217-240, March 2006.
- Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiawicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu. OpenDHT: A Public DHT Service and Its Uses. *ACM SIGCOMM*, August 2005.
- P. Brighten Godfrey and Ion Stoica. Heterogeneity and Load Balance in Distributed Hash Tables. *IEEE INFOCOM*, March 2005.
- P. Brighten Godfrey and David Ratajczak. Naps: Scalable, Robust Topology Management in Wireless Ad Hoc Networks. *Information Processing in Sensor Networks (IPSN)*, April 2004.
- Brighten Godfrey, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp, and Ion Stoica. Load Balancing in Dynamic Structured P2P Systems. *IEEE INFOCOM*, March 2004.
- Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, Trees and Minimum Latency Tours. *FOCS 2003*: 36-45.

Papers under submission:

- P. Brighten Godfrey, Matthew Caesar, Ian Haken, Yaron Singer, Scott Shenker, and Ion Stoica. Stabilizing BGP, Safely.

Talks

January 2008	“Balls and Bins with Structure”, SODA 2008
July 2007	“Stable Internet Route Selection”, Tech Talk, Cisco Systems
June 2007	“Stable Internet Route Selection”, NANOG 40
September 2006	“Minimizing Churn in Distributed Systems”, SIGCOMM 2006
August 2006	“Minimizing Churn in Distributed Systems”, SDI/LCS Seminar Series, Carnegie Mellon University
July 2006	“On the Price of Heterogeneity in Parallel Systems”, SPAA 2006
March 2005	“Heterogeneity and Load Balance in Distributed Hash Tables”, INFOCOM 2005
April 2004	“Naps: Scalable, Robust Topology Management in Wireless Ad Hoc Networks”, IPSN 2004
January 2002	“Text Oriented CAPTCHAs”, First Workshop on Human Interactive Proofs, Xerox PARC

Teaching and Advising Experience

- October 2007 **Guest Lecture**, “Interdomain Routing”, EE122 (Undergraduate networking course; Instructor: Prof. Vern Paxson)
- September
2006-present **Advising** of undergraduate, Ian Haken, on research projects
- January-May
2006 **Private tutor** for a student in masters-level course CSC810.01, “Analysis of Algorithms II”. Instructor: Prof. James Wong, San Francisco State University
- January-May
2005 **Teaching assistant** for CS 270, “Combinatorial Algorithms and Data Structures”. Instructor: Prof. Christos Papadimitriou, University of California at Berkeley
- August-
December
2004 **Teaching assistant** for CS 61B, “Data Structures and Advanced Programming”. Instructor: Prof. Paul Hilfinger, University of California at Berkeley

Other Professional and Community Activities

- **Reviewer**, *Conferences*: EuroPar (2006), ACM SIGCOMM (2005, 2004), IPTPS (2005), HotNets (2005), ANCS (2005), NSDI (2004), DISC (2003), TRIDENTCOM (2008); *Journals*: IEEE/ACM Transactions on Networking (2007, 2006), Operations Research (2006, 2005, 2004), IEEE Transactions on Parallel and Distributed Systems (2007, 2006), Elsevier Journal on Ad Hoc Networks (2007, 2005), Elsevier Computer Networks (2006), Elsevier Performance Evaluation (2004), and others.
- **Compiled and maintained** a Repository of Availability Traces: Traces from measurement studies of a number of distributed systems, packaged in a common compact format (see <http://www.cs.berkeley.edu/~pbg/availability/>).
- **Member**, CS Graduate Student Association Faculty Candidate Evaluation Committee, 2007.
- **Student member**, graduate student admissions committee in Systems area, 2005.
- **Co-organizer**, International House Debate Club, 2004-2006 (see my article in the International House Times, Spring 2006, p. 3: <http://ihouse.berkeley.edu/a/times/spring2006.pdf>)

References

- **Prof. Ion Stoica**
Associate Professor, Computer Science Division
University of California at Berkeley
Berkeley, CA 94720-1776
510-643-4007
istoica@cs.berkeley.edu
- **Prof. Scott Shenker**
Professor, Computer Science Division
University of California at Berkeley
Berkeley, CA 94720-1776
510-666-2880
shenker@icsi.berkeley.edu
- **Prof. Richard M. Karp**
University Professor, Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720-1776
510-642-4274 x 122
karp@cs.berkeley.edu
- **Prof. Emin Gün Sirer**
Associate Professor, Computer Science Department
Cornell University
Ithaca, NY 14853
607-255-7673
egs@systems.cs.cornell.edu

Research Statement

Philip Brighten Godfrey

February 10, 2008

In my research I am excited to be able to combine the *practical design of networked systems*, such as the Internet and overlay networks, with the use of *theoretical analysis* to expose principles fundamental to many systems.

Previous work

My dissertation addressed systems design in heterogeneous environments. Parallel and distributed systems have become increasingly heterogeneous in recent years. Rather than running on clusters or supercomputers composed of identical nodes, many modern distributed applications—including peer-to-peer systems, grid computing, applications running on platforms like PlanetLab, and the Internet itself—use nodes which span the world and are administered by different entities. As a result, these nodes differ in many dimensions, such as available bandwidth, processor speed, disk capacity, security, and reliability. Even within a data center, nodes are not identical since upgrades are performed in installments.

A theme of my dissertation is that heterogeneity can not only be handled, but rather should be viewed as an asset. I developed practical methods for specific systems to adapt to and take advantage of heterogeneity (such as avoiding failures in Internet routing) as well as principles that can be applied to many systems (such as the fact that randomizing node selection typically reduces turnover in the population). My results show how performance and reliability can be improved in heterogeneous environments.

One of the hardest and most fundamental problems in networking is Internet routing. For roughly two decades, the flow of data on the Internet has been directed by the Border Gateway Protocol (BGP), a path vector routing protocol which allows separate administrative entities to offer paths to each other and to select paths along which to route. I addressed how to deal with *churn* in BGP—that is, failure and replacement of routes—which is known to cause periods of outage in the data plane and significant CPU utilization on core routers. Despite the fact that the problem of churn was recognized more than a decade ago, there currently is no compelling mechanism for stabilizing BGP routes. The principal technique, route flap damping, is widely deployed but is now recognized as counterproductive because it delays route convergence and sacrifices availability. With concerns about BGP’s scalability recently prompting a renewed interest in stability, we pursued a more principled approach to stabilizing Internet routing. In particular, using both lower and upper bounds, we characterized the feasible points in the tradeoff spaces between *stability*, *availability*, and the degree of *deviation* from the path preferences of the network operator. For example, one of our lower bounds shows that within our large-scale evaluation environment driven by traces of failures on the Internet and under certain assumptions, stability cannot be improved by more than about $8.1\times$ unless we either sacrifice availability or violate customer-provider-peer routing relationships. Our upper bounds include new *Stable Route Selection (SRS)* strategies. SRS prefers more stable paths when there is a choice, relying on heterogeneity among the choices in order to reduce churn. Simulations using real-world data, complemented by software router experiments, show that SRS preserves the high availability of BGP without flap damping while obtaining slightly better stability than BGP *with* flap damping, and coming within $1.6\times$ of our theoretical lower bound. Alternately, SRS can trade off stability for less deviation from preferred paths. These results demonstrate both what *is* and what *isn't* possible with regard to stabilizing BGP.

Selection of routes in BGP is just one example from among a wide variety of systems that need to select components to use, like routes or nodes, from among a heterogeneous pool of available components. We studied how to minimize churn using node selection strategies applicable to many distributed systems. A key result drew a distinction between two strategies. *Random Replacement (RR)* picks a uniform-random available node when a new or replacement node is required in the system. What we call a *Preference List (PL) strategy* ranks the nodes according to some fixed preference ordering unrelated to churn, and picks the most preferred available node. Although both RR and PL pick nodes in a way apparently unrelated to reliability, we found a surprising difference in their behavior. While PL

strategies perform poorly with respect to churn, RR is quite good—typically within a factor of 2 of the performance of heuristics like Longest Uptime (LU) that intentionally select reliable nodes! This effect persisted across a diverse collection of synthetic and real node failure traces. In a stochastic model, we explained the effect, which stems ultimately from the heterogeneity of the nodes’ session time distributions, and a statistical effect known as the inspection paradox.

These results are significant for two reasons. First, understanding RR and PL is useful in understanding the numerous real systems in which they appear. For example, in a multicast tree construction scenario studied, we explained the effects of random parent selection that were only partially explained in past work. Examples of PL strategies abound as a result of optimizing for objectives other than churn: a client picking the closest server, for instance, or BGP’s selection of routes based on local preference or on routers’ IP addresses. Second, RR’s simplicity and robustness to misbehavior can make it a preferred choice. For example, randomizing the Chord distributed hash table’s finger selection—a trivial change to its design—cut the churn-induced end-to-end packet loss rate by 29% in a real-world failure pattern. In contrast, the LU strategy would require querying many nodes to request their uptime and trusting that the answers are not malicious, or continually probing many nodes to directly monitor their uptime.

In addition to *reliability* heterogeneity addressed in the above work, my dissertation dealt with systems design under heterogeneous *node capacities* such as bandwidth, CPU cycles, and disk storage. I addressed this problem for distributed hash tables (DHTs), which provide a hash table-like substrate that has been used to build services such as BitTorrent’s distributed tracker. Early DHTs were designed primarily for equal-capacity nodes. Our design, called Y_0 , flexibly and provably adapts to any capacity distribution, obtaining significantly reduced overhead and shortened route lengths when nodes are heterogeneous, as well as a better load balance. Y_0 ’s techniques also inspired later work in which I showed that in the balls-and-bins model of balanced allocations, a ball’s choices of bins can be correlated in a very general way as long as each ball has $\Omega(\log n)$ choices.

A common thread unifies my work: we observed better stability, lower route lengths, better load balance, or lower overhead in *heterogeneous* environments than in *homogeneous* ones. But clearly there are some situations where more heterogeneity is detrimental, and others where heterogeneity helps; how general is the latter case? We formalized that question with a framework, the *price of heterogeneity*, and within it delineated a large class of models of systems in which increasing capacity heterogeneity can never be much of a disadvantage. This class included job scheduling problems, models of load balancing in DHTs, and network degree/diameter tradeoffs. In an extension of the price of heterogeneity, we showed that under certain technical assumptions, RR’s churn decreases as node session times become more heterogeneous.

Future work

I see *adaptability*—dynamic reaction to diverse operational environments—as a major challenge for computer systems over the next decade. Pressures are coming from several directions: as computer networks and systems become a greater part of society, they operate in a wider variety of environments and interact with increasingly complex systems. At the same time, we desire better adaptation to the human users in system’s environment, and we desire greater dependability, which may require adapting to many different failure conditions. My dissertation work on adapting to heterogeneous failure patterns and node capacities addressed aspects of this problem, but much work remains.

Three key requirements to facilitate adaptability are *obtaining feedback* from the environment, building on top of a *flexible infrastructure* that permits many possible actions, and using *adaptive algorithms* which respond to the feedback within the flexibility afforded by the infrastructure. The current Internet is deficient in all three areas, and I plan to address these problems.

Feedback and flexibility. Automatic routing decisions in today’s Internet are largely decoupled from data plane objectives like load balance, latency, and end-to-end availability. As a result, the feedback loop goes through humans: as much as Internet routing is automatic, it can also be said to be manual, with operators across the globe tweaking inputs to the BGP decision process to achieve desired traffic engineering or policy effects. This arrangement has a significant cost in human time, and neglects the useful information available to the two endpoints in a connection.

The Internet’s routing infrastructure is also extremely inflexible. End-hosts have no choice in the paths their packets travel, and routers choose among only a fraction of the possible policy-compliant paths. What flexibility does exist is limited further since adaptive automatic decisions can interfere with manual configuration. Network operators have related to me that this problem caused many operators to turn off stability features implemented by Cisco and Juniper.

A solution to the problems of feedback and flexibility is to give end hosts (or their representatives, edge routers)

some amount of control over their packets' routes. This gives flexibility to the entities that have access to feedback, potentially yielding huge benefits in reliability and performance. Indeed, limited source routing or negotiation of alternate paths is a feature of many proposals. But how close can we come to exposing to end hosts the full diversity of available policy-compliant paths, while still giving network providers sufficient control over their own networks, and allowing the system to scale? An approach I plan to study is to allow source routing over short "pathlets" which are advertised in accordance with network providers' policies.

Adaptive algorithms. With new routing architectures come new opportunities and challenges for the design of adaptive routing algorithms. In particular, source routing such as the "pathlet" approach shifts the burden of failure detection and traffic engineering onto the end-hosts. I am interested in leveraging online learning algorithms to select near-optimal routes, potentially with the help of collaboration among end-hosts or routers to share learned information.

In addition, it is important to do as much as we can with the limited flexibility of the present-day routing architecture. I believe my recent work on Stable Route Selection, when used in a mode with little deviation from preferred paths, holds promise for a practical, safe replacement for route flap damping and is deployable in the current Internet. I plan to engage with industry contacts to work towards that goal. Over the past six months I have begun forming relationships by presenting talks at a North American Network Operators' Group (NANOG) meeting and at Cisco, and by co-authoring a grant proposal that was recently funded by the Cisco Collaborative Research Initiative. Motivated by our lower bounds which suggest that a dramatic improvement in stability is impossible within BGP, I am also exploring a scheme which requires small changes to BGP but which could reduce churn much more significantly by localizing path change announcements.

As a next-generation Internet architecture will be expected to support decades of growth and novel applications, it is particularly important that architectural choices be based on a solid foundation. Thus, I believe the above problems are prime candidates for the flavor of theory-informed systems design that I have successfully employed in my past work, providing guarantees of the behavior of a proposed design, as well as an understanding of what goals and tradeoffs are and aren't achievable.

Teaching Statement

Philip Brighten Godfrey

January 17, 2008

Teaching is exciting for me and is critically important for the discipline of computer science: for generating versatile new computer programmers, for generating new research ideas, and for generating new researchers. With this in mind, I want to emphasize several themes that guide my teaching.

- **Challenging the status quo.** Learning an idea is useful; learning how to learn ideas is better. I want students to go a step farther and *learn how to learn ideas that no one else has learned before*—that is, how to innovate. Students should be continually encouraged to skeptically challenge the status quo, to imagine what *might* exist. As an example, for decades Internet communications have almost exclusively used either TCP (a reliable stream-based protocol that controls congestion) or UDP (an unreliable datagram protocol that ignores congestion). I would feel a moment of success if my students ask, why should these particular combinations of features be the only two choices? Why not, say, an unreliable yet congestion-controlling protocol? Actually, there is good reason for such a protocol for real-time applications, and it is the subject of recent work.¹ My teaching will be geared towards provoking these questions. I will ask for nonstandard solutions to real problems, and I will emphasize recent research advances in order to teach innovation by example and keep material fresh.
- **Interactivity.** Interaction among students and between the teacher and students helps all of us learn from each other and stay interested. Lectures should have a strong interactive component. In an undergraduate data structures class, I sometimes had students execute algorithms “live” in recitation, such as executing merge sort with each person performing one recursive call. Interactivity can go beyond the lecture, too. One idea appropriate for a networking course is to have students build a software router and connect the students’ routers together in an overlay network. Evaluation of each router would then be based in part on the number of *other* students’ routers with which it could maintain an end-to-end flow, with live statistics on the course web page. This would encourage students to collaborate on getting their routers to speak with each other successfully.
- **Conveying ideas effectively.** This is probably the most basic task in teaching and depends on two key elements, both of which I especially enjoy. One is figuring out how to explain a concept in multiple ways “on the fly” during a lecture or while working with students one-on-one. The second key element is getting feedback on what explanations worked or didn’t work, why, and for whom. This involves personal sensitivity to students, and is facilitated by having multiple channels through which to receive feedback. One channel which I found very helpful was to accept anonymous questions and comments on note cards after discussion sections. Another channel I have used is more empirical: when presenting my work on “Minimizing Churn in Distributed Systems” in a graduate course at Berkeley, I included three separate explanations for why simply randomizing node selection reduces churn, and polled the audience on which one made the most sense. One of the three came out the clear winner, which showed me which explanation was effective and which needed to be revised or dropped. The more traditional channels of accepting feedback during lectures, office hours, and by email are also critical.

At Berkeley, I had the opportunity to be a teaching assistant for an undergraduate data structures course, including a discussion section and a lab, and for a graduate algorithms course. My teaching evaluations in these courses gave me overall scores of 4.4/5 and 4.3/5, respectively, compared with a mean of 3.9 in both courses for all TAs during the period that records were kept. I have also worked as a private tutor. Over the past 18 months, I have gained experience leading a research project, for which I am coordinating the activities of an undergraduate (who I advise), a first-year graduate student, and a recent graduate who is a postdoc at another institution. My roles outside academia have also helped me learn how to lead a group: for two years at Berkeley, I organized a debate club that presented several events

¹See Kohler, Handley, and Floyd, “Designing DCCP: Congestion Control Without Reliability”, SIGCOMM 2006.

each semester; I am co-leader of an actively gigging seven-piece brass band; and I have previously led several jazz combos.

I am prepared to teach core graduate or undergraduate courses in Networking, Distributed Systems, Algorithms, and Operating Systems, as well as any introductory computer science course. I also plan to develop courses or seminars covering such current research areas as dealing with failures in networked systems, overlay and peer-to-peer networks, and Internet routing. The latter would take a broad view of the topic, including a traditional treatment of Internet routing protocols but ranging from theoretical aspects (such as recent results concerning the complexity of BGP convergence) to operational aspects (such as how ISPs do traffic engineering in the real world). This would set the stage for readings on new Internet architecture proposals.

Given my experiences at Berkeley, I believe I am well prepared to teach, advise students, and lead a research group—and I am certainly excited to get started!