

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.