

Research Statement

Richard P. Martin

Department of Computer Science
Rutgers University
August 2005

1 Introduction

My work has spanned a seeming widely diverse set of topics, covering radio-based localization, density control in sensor networks, availability for Internet services, peer-to-peer (P2P) systems, and high-performance networking. Although these topics seem completely disjoint, they are all examples of *local computer systems*. Briefly, such a computer system is comprised of a set of components that must take actions with incomplete information. I will expand on my definition of that term in Section 7.

Regardless of the specific topic, my work has consistently used a combination of science and engineering grounded on theoretical foundations. With the goal of impacting the practice of computer science, my work requires design, experimentation, quantitative and qualitative evaluation, analytic modeling and simulation to answer questions about computer systems and methodologies related to their application.

My work has been funded by a Defense Advanced Research Projects Agency (DARPA) grant, 4 National Science Foundation (NSF) grants, including a CAREER award, as well as grants and equipment donations from IBM and Cisco. To date, these efforts have lead to 2 students expecting to complete their Ph.D.s in May 2006, 3 additional ongoing Ph.D. students, and 4 graduated master's students, along with several undergraduate research projects. The details are listed in my C.V.

In the remainder of this statement, I will first describe the motivation, current and future work for my localization research in Section 2. In Section 3, I briefly describe my work on sensor network density. Section 4 describes my availability research, and Sections 5 and 6 describe my work in P2P and high performance networks. In Section 7, I conclude with a definition of computer systems and a description of my style in conducting this research.

2 Radio-based Localization

I began my localization work early in 2004 with Ph.D. students Xiaoyan Li and Eiman Elnahrawy as a single Principal Investigator (PI) project. Since that time, I have also started work with 2 new Ph.D. students, John-Austen Francisco and Yingying Chen, as well as collaborations with Dr. David Madigan and Dr. Wade Trappe in the Rutgers Statistics and Electrical and Computer Engineering departments, respectively. In the summer of 2005, I joined the Faculty at the Rutgers WINLAB and began collaborations there as well. Radio-based localization will remain a main focus of my research until we have fully explored all its impacts and limitations.

Motivation. The primary goal of this research is to provide a scalable, general purpose localization infrastructure necessary to integrate location information into any computing device with a radio. My aim is to enable localization to become similar to communication, which is now integrated into nearly every computing device. Advances in Moore's law (i.e., the doubling of a device's transistor count every 2 years) are causing declining costs of radios, and today they are cheap enough to include in many everyday devices, e.g., keyboards and mice. Given the predicted even lower cost of radios, a radio-based localization has the potential to have a high impact in the way we think about how to find and use many kinds of objects.

In order to be general purpose, my work attempts to leverage the existing communication infrastructure to provide positioning. Reusing the infrastructure would provide a tremendous cost and deployment savings over a specific localization infrastructure, such as ceiling-based ultrasound.

A general purpose localization system would enable a host of novel monitoring, tracking, routing and security services. For example, inventory tracking is currently limited to highly structured manufacturing and shipping chains; my research would make feasible much more ad hoc types of applications. For example, a challenging application would allow parents to monitor their children in a shopping mall. A less robust version of such a system would allow a professor to localize books taken by students. Security is another area where our algorithms could be applied, for example, a user could be paged when objects enter or leave authorized areas.

Current Work. A general purpose localization system should manage the uncertainty in the position estimation. This is especially true in indoor environments because the signal propagation deviates significantly from free-space models. The resulting noise, and especially bias, make accurate localization challenging.

To describe uncertainty we began by developing area-based algorithms [22]. These algorithms can both represent the set of possible locations of an object as an area or volume in addition to returning a single location. We developed 2 sample algorithms and several new metrics that are applicable to any area-based approach, giving mathematical rigor to the concepts of localization accuracy and precision. We found that area-based approaches are better able to utilize and describe uncertainty in a meaningful manner as compared to point-based approaches. For example, area-based approaches can direct a user's search for an object when the position estimate has a high error.

We then characterized the limits to localization using signal strength in indoor environments [21]. We showed that although a broad spectrum of algorithms can trade-off accuracy for precision, none has a significant advantage in localization performance. We also presented strong evidence that these limitations are fundamental and that they are unlikely to be transcended without highly detailed environmental models or additional localization infrastructure.

Next, we explored a novel use of Bayesian networks for localization [48]. Much like our other area-based algorithms, a desirable property of Bayesian networks is that they scale gracefully under a variety of challenging environments, such as those with degraded signal quality and environmental distortions. A key contribution of this work is that we found that certain Bayesian networks can localize radios with *no training data*, i.e., data collected at known locations to assist with later localizations. Bayesian networks can thus localize by observing the existing packet traffic, which is a qualitative improvement over current learning methods.

Another contribution of our Bayesian approach is its generality: it can take advantage of different hardware, software, and prior knowledge within a single probabilistic framework. For example, we have successfully applied our work to laptops using 802.11 hardware, as well as Motes using 802.15.4. We have also shown how to scale the Bayesian networks to use the information from additional infrastructure over the basic signal strength. For example, we have used signal-strength in combination with directional antennas to take advantage of angle-of-arrival information [20].

More recently, we have begun to characterize the underlying signal bias and noise in indoor environments [42]. We developed a ray-sector model that constructs signal distortion maps by adding signal bias within sectors defined by rings and randomized rays using a traditional log-linear decay model as a baseline. We showed our model generates shapes similar to those in measured signal maps. We also demonstrated the utility of our model for a higher-level application by showing it accurately predicts the localization performance for two dissimilar localization algorithms.

Related Work. The last 5 years have seen intense work on localization systems for wireless LANs, ubiquitous computing, and sensor networks, and it is impossible to cover everything in this small section. Rather, I will only give examples to illustrate the similarities and differences to our research. Our approach is distinct in several ways from much of the localization work in sensor networks because those goals mostly concern positioning the sensors and detected phenomena. These often lead to using infrastructure beyond radios, such as infrared, sound, ultra-sound or optics [33, 30, 63, 66, 75]. In addition, line-of-sight to peers or infrastructure is often

assumed, especially in outdoor settings, although our work makes no such assumption.

Our work also is distinct from the large body of recent work on aggregate-based localization [18, 34, 59]. There, the goal is to use a group of sensors, coupled with limited or inaccurate information about each sensor, and use the resulting over-specification generated by a group of sensors to localize them. Another set of works employing aggregate information are those using collaborative signal and information processing, which can, for example, localize targets [79]. To date, I have focused on the information we can extract from the infrastructure because this simplifies many problems and reduces concerns arising from administrative constraints found in indoor environments (e.g., not every device is under centralized control and willing to cooperate).

This work is most related to the many positioning systems that use machine learning techniques, such as nearest neighbor[3], support vector machines [5], and Bayes' rule [78], in combination with observations about signal strength to localize radios. A critical distinction from these works is that our algorithms use areas to describe position uncertainty. We also are focusing on algorithms that minimize or eliminate the amount of training data. Training data is very expensive to collect, because it requires either sending people to record radio signal parameters at known locations, or adding many extra devices (sniffers) to the infrastructure.

Future Work. We are currently addressing localization accuracy by adding more hardware to the infrastructure, as well as improving the algorithms. On the hardware side, we have added directional antennas [20]. We are also pursuing adding Time-of-Arrival (TOA) information to incoming packets and then using a radio interferometry approach (similar to what astronomers use). However, getting TOA information is quite a challenge and this will take several months.

On the algorithmic side, we found that our current network is too sensitive to signal-to-distance bias. We are thus creating new Bayesian networks that weight the angle information more heavily than the distance information when the signals are below a threshold. We are also embarking on replacing our current Monte-Carlo techniques with variational approximations to speed the localization latency. Finally, we are investigating using our ray-sector model to generate realistic connectivity topologies as compared the circular propagation models typically used.

I am also engaging in research on localization security [11]. We have recently built and experimentally verified a simple attenuation based-attack model. Next, we will explore the robustness of algorithms to these attacks. Our preliminary results show that area-based algorithms are more robust than point-based ones, because they scale better under the increasing uncertainty during an attack. We also found that the Bayesian networks approach was qualitatively more robust than the other algorithms. However, we have yet to ground these empirical results on theoretical foundations.

3 Managing Density in Sensor Networks

I began my density control work in the fall of 2002 with Xiaoyan Li, and Andrew Tjang, Ph.D. students, as well as Dr. Thu Nguyen and Dr. Badri Nath in the Computer Science department. I am no longer pursuing this topic.

Motivation. The motivation for this work centers on the observation that if very low-cost (e.g. less than \$10) sensor nodes become available, very high network densities will result. To quantify this claim, we recorded the smallest bounding box in 3-space of all 527 objects in one of our laboratories. If each object were tagged with a radio, then even a small radio range of 1m generates a large variance in the out-degree (the number of reachable nodes), spanning from 2 to 128.

Results and Contributions. Some initial exploratory work in this area examined the impact of mobility on diffusion routing algorithms [13]. The range control work was an analytic model and algorithm to control the radio range (how far a radio signal can travel) to optimize 1-hop broadcast messages, i.e., the number of nodes that will probabilistically receive a 1-hop message [44, 45]. The contribution of this work is that we can optimize 1-hop broadcast using only the locally observed sending rate and node density. We showed how a node running our algorithm can observe these parameters using only message eavesdropping and thus does not require extra

protocol messages. We demonstrated using simulation that in spite of many simplifications in the model and incomplete density information in a live network, our algorithm converges fairly quickly and provides good coverage for both uniform and non-uniform networks across a wide range of conditions.

The second direction examining density grew from the observation that under high node densities, a bus based sensor network organization has many advantages over a traditional radio based sensor network. This observation led to our work on Active Tapes, which are both communication and power bus organizations for sensor nodes [70]. We constructed simple cost models to compare active tapes against more traditional wireless sensor networks. We found that density, lifetime, and power consumption play significant roles in determining overall deployment and maintenance cost. We then characterized regimes where tapes are always cost effective, sometimes cost effective, and never cost effective. We described three real-world applications in terms of our cost models. We also built a prototype implementation of an active tape for power distribution and networking that both validated our models and is used by Dr. Ulrich Kremer's EEL laboratory at Rutgers.

4 Computer System Availability

I began research addressing computer availability in 2001 with a Samian Kaur a Masters student, Taliver Heath and Kiran Nagaraja, both Ph.D. Students, and Dr. Thu Nguyen. In 2002, I began collaborating with Dr. Ricardo Bianchini and several additional students exploring performability analysis. That same year I also began our compiler-based testing work with Chen Fu, Ph.D. student, and Dr. Barbara Ryder from Rutgers, and Dr. David Wonnacott from Haverford College. These are all very close collaborations, with everyone filling different roles as needs arise. Availability will remain a focus of my research for a long time because this is a fundamental, and interesting problem, that will require many efforts by all areas of computer science.

Motivation. As computers permeate all aspects of our lives, a wide range of computer systems must achieve high dependability. This broad concern includes a wide range of issues, such as availability, integrity, maintainability, and security. Unfortunately, even limiting our concern to availability (the percentage time a system delivers correct service), few computer systems can legitimately claim to be highly available, e.g., 99.999%, or 5 minutes of unavailability per year. Indeed, dependability issues continue to make front page news.

Many studies conducted over the past 20 years have empirically observed that the two largest contributors to system unavailability are software faults and human mistakes [31, 53, 58, 61]. The studies show that in order to improve system availability, designers and operators will need to use a broad array of techniques to avoid, tolerate, and repair software faults and human mistakes. My availability research has thus focused on three areas: (1) performability analysis (a combined metric of performance and availability), which is needed to quantify our results, (2) software testing to improve system software, and (3) human-aware system design, to reduce or avoid the impact of human mistakes.

I believe that advances in computer systems availability will come in smaller increments, i.e., there will be no silver bullet. Successful research will become incorporated into a constellation of known techniques used to design, evaluate and operate highly dependable systems. Measurement, instrumentation, and modeling will be especially important because practitioners often do not have the time and resources to quantify availability. New methodologies can also have important impacts, because today the cost of adding dependability to systems is still quite large.

Performability Analysis. An early line of work characterized the reboot behavior of workstations in a cluster [36]. That initial analysis led to later analytic models that predict the behavior of workstation reboots (a recently rebooted machine is likely to reboot again). We incorporated these models into algorithms that increase overall service availability by better provisioning and scheduling of workloads on a cluster [37].

We explored systematically evaluating and improving the performability of Internet Services. A key contribution of our work is a two-phase methodology for quantifying performability [55]. We have successfully applied this methodology to a wide range of services and servers, thus demonstrating its generality. In the first phase, evaluators use a fault-injection infrastructure to characterize the server's behavior in the presence of faults. In the

second phase, evaluators use an analytical model to combine an expected fault load with measurements from the first phase to assess the service’s performability. Using this model, evaluators can study the server’s sensitivity to different design decisions and fault rates.

To demonstrate our methodology, we first built a fault injection infrastructure[43], which was successful transferred to the Federal University of Minas Gerais in Brazil and the Recovery Oriented Computing (ROC) group at U.C. Berkeley [8]. We first applied our methodology to 4 versions of the PRESS Web server using 7 classes of faults, quantifying the effects of different design decisions and assumed parameters of the target environment on performance and availability[57]. We also applied this methodology to evaluate the performability of numerous designs for a range of sub-systems: the communication system [56], a set of fault-mapping strategies called fault model enforcement (FME) [54], and various state management strategies for user data [28].

Software Testing. We developed a new approach that uses compiler-directed fault-injection for coverage testing of recovery code in Internet services to evaluate their robustness to operating system and I/O hardware faults [27]. We first defined a new set of program-fault coverage metrics that enable quantification of Java `catch` blocks exercised during fault-injection experiments. Next, we used compiler analyses to instrument application code in two ways: to direct fault injection to occur at appropriate points during execution, and to measure the resulting coverage. To prove these ideas we applied our techniques manually to *Muffin*, a proxy server; we obtained a high degree of `catch` block coverage, with on average, 85% of the expected faults per `catch` being experienced as caught exceptions.

Human Factors Impacting Availability. Human mistakes are so common and harmful because computer system designers have consistently failed to consider the human-system interaction explicitly. Given observed failure rates caused by human errors, the effect of humans on overall dependability should become a first-class design concern in complex computer systems[6]. More specifically, we believe that human-factors studies are necessary to identify and understand the causes, types, and likelihoods of human mistakes. By understanding human-system interactions, designers will be able to build systems to avoid, hide, or tolerate these mistakes, resulting in significant advances in dependability.

We have thus begun a novel research direction in computer systems that uses human factors studies to both quantitatively and qualitatively understand how human mistakes impact availability [58]. In that work, we first characterized operator mistakes in Internet Services by performing an extensive set of experiments using 21 human subjects performing tasks on a realistic three-tier auction service. We observed 42 mistakes ranging from software misconfiguration, to fault misdiagnosis, to incorrect software restarts. To reduce the impact of these mistakes, we developed a *validation* sub-system for operator actions that hides mistakes from the rest of the system. We demonstrated how to accomplish validation using a prototype that is an extension of the online system. Our approach allows components to be validated using both real and synthetic workloads before they migrate back into the running service. We found our validation effective, detecting 66% of the operator mistakes in our experimental data.

Related Work. There has been extensive work on analyzing faults and how they impact systems [32, 69, 41], as well as studies benchmarking system behavior under fault loads [38, 46]. These works differ from ours because they do not provide a good understanding of how one would estimate overall system availability under their fault loads. There have also been a large number of system availability studies. The two approaches that are used most often include empirical measurements of actual fault rates [2, 37, 47, 39, 53] and a rich set of stochastic process models that describe system dependencies, fault likelihoods over time, and performance, e.g., [29, 52, 67].

It is interesting to compare our performability modeling approach, which is based on a simple 7-stage piecewise linear template, with the traditional modeling approaches using stochastic processes [52, 67]. Our template describes the server’s response to a fault as a linear sequence of transitions with a deterministic time per state. This approach makes modeling much simpler compared the task of quantifying the states, probabilities and transitions in a stochastic model. As a result of using the template with deterministic transitions, we can use

simple algebraic techniques to solve for the average availability and reward (throughput in our case). In essence, we have traded some generality for ease of measurement, model parameterization and complexity. We have found both the template and approach fairly general, as they can adequately describe tradeoffs in a clustered 3-tier on-line bookstore, a 3-tier online auction, a database, and the PRESS web server. The model makes a good trade-off between limiting generality to a specific important domain and simplicity for practical application.

Our testing experiments most closely resemble those measuring responses to errors using traditional program-coverage metrics, notably [71]. That work placed breakpoints at key program points along known execution paths and injected faults at each point, (e.g., by corrupting a value in a register). Their work differs from ours in its goal, the kinds of faults injected, and their definition of coverage.

Our human factors experiments are most closely related to recent ones examining human interactions with disk arrays [9] and databases [4]. In our studies, however, we performed experiments on wider range of users (from graduate students to professional operators) and tasks, such as adding hardware and upgrading software. In addition, we showed how to extend the system with validation to hide many observed mistakes. A work with similar goals to validation is a system for allowing operators to undo actions [10] caused by mistakes. That work is complementary to ours because their strategy is to reduce the time to repair, while validation tries to hide mistakes instead. Another related area that addresses human mistakes is autonomic computing [40]. Regardless of how successful the autonomic computing is, humans will always be part of the installation and management of complex computer systems at some level. In the foreseeable future, humans will continue to be responsible for determining a system's overall policies, as well as for addressing unexpected behaviors and failures. Thus, some form of human mistakes will be inevitable.

Future Work. One of our immediate goals is to show our work can include traditional databases. Towards this end, we have characterized the typical administrator tasks, testing environments, and mistakes, using results from an extensive survey we have conducted of 51 experienced administrators. We have also begun experimentation with a validation strategy for database administrator actions. Our preliminary results show that validation can detect the major classes of administrator mistakes [60].

We are also investigating methods to detect human mistakes via modeling using 2 strategies. The first is to use some of the performance models we have developed and apply them to signal when an operator action may adversely impact system performance, which should help avoid mistakes. The second approach is to allow systems designers to specify correct system behavior using a special assertion language called **A**. The **A** runtime executes assertions against a set of observations of a live system. We currently have an initial **A** compiler and runtime and have written small programs for a distributed web server. We plan to expand the programs to include larger services as well as evaluate various **A** programs' effectiveness at reducing the time to repair needed under component faults and operator mistake scenarios.

Finally, in the human factors realm, we will address the limitations caused by the short duration of our current experiments by running ones that last for weeks and months. The hours long time-scales of our previous experiments meant that we did not account for a host of effects that are difficult to observe at these time-scales. For example, the effect of increasing familiarity with the system, the impact of user expectations, systolic load variations, stress and fatigue, multi-fault scenarios, very complex tasks, and the impact of system evolution as features are added and removed.

5 Peer to Peer Computing

I began participating in this work in 2002 working with Matias Cuenca-Acuna and Chris Peery, both Ph.D. students, as well as Dr. Thu Nguyen, who leads this effort. In 2004, Konstantinos Kleisouris, a Ph.D. student, also began working with me on this topic. I am no longer pursuing this research area.

Randomized Gossiping for Search and Retrieval. Our work investigated how to construct document search and retrieval (e.g. a search engine) in a P2P environment. The prototype realizing our work is called PlanetP [15]. The main challenge is how to virtualize the search in a completely distributed manner while the peer membership

has a high degree of variability. PlanetP is unique in two ways: (1) it used gossiping (an information exchange with a randomly selected peer) to globally replicate peer membership and the content index, and (2) it used a completely distributed search and ranking algorithm, because ranking is a critical function of document stores. Using simulation and a prototype implementation, we showed that PlanetP achieves ranking accuracy that is comparable to a centralized solution and scales easily to several thousand peers while remaining resilient to changing membership.

Probabilistic Availability Using Gossiping. A critical question left open in the first PlanetP prototype was how to manage the availability of unpopular documents when many peers often migrate between offline and online states. We thus explored how to improve the availability of shared data for P2P communities where individuals may be disconnected often and for extended periods [14]. In particular, we addressed the question of how to place replicas of shared files in such a way that, despite constant changes to the online membership, the files are highly available without requiring the continual movement of the replicas. Our algorithm uses randomized decisions extensively together with a novel application of an erasure code to tolerate autonomous peer actions as well as staleness in the loosely synchronized global state. We showed that while peers act autonomously, the community as a whole will converge to a stable configuration. We also showed that storage is used fairly and efficiently, delivering 99.9% file availability at a cost of six times the storage footprint of the file collection when the average peer availability is only 24%.

Related Work. We targeted communities of thousands of peers because most other efforts have ignored this range in attempting to scale to millions of users [65, 68, 80]. Our target range can have significant impact: there are currently many communities around this size such as those served by Yahoo Groups, and Dalnet's IRC servers. On a different front, our gossiping approach can also be applied to manage distributed computing environments such as grid systems (e.g., maintaining membership, service description, and aggregate statistics); recent work shows the promise of such a P2P management approach [25]. Thus, our work explored the question of whether certain functionalities, such as content ranking, that are extremely difficult to implement in very large P2P systems becomes possible to implement at our target scale.

6 High Performance Networking

I began working on high performance networking in 1993 under the supervision of my Ph.D. advisor, Dr. David Culler at U.C. Berkeley. At Berkeley, I worked with many other graduate students and professors in this area. Our Active Messages code was incorporated into the Inktomi Search engine (which was acquired by Yahoo!) in 1996. In 1998, I also collaborated with Marc Ficuzynski and Dr. Brian Bershad at the University of Washington. I continued pursuing this area after I joined Rutgers, collaborating with Samian Kaur, Taliver Heath and Dr. Thu Nguyen. My last work in this area was in 2001.

Active Messages. My initial research concerned the development of low latency, high bandwidth communication on commodity workstations. The goal was to enable a new range of parallel programs and systems to run on Network of Workstations (NOW) platforms by greatly improving the communication performance.

I first developed a version of Active Messages [73] on HP workstations [49], and a second version for Sun workstations that was later used by Inktomi. Both these experiences taught me important lessons about the relationship of research to practice, which I will describe more in Section 7. We were successful in our goal to demonstrate practical high performance communication commodity hardware and operating systems. Indeed, it was impressive to see 4 HP workstations run parallel codes as fast as a 32 processor CM5. We also successfully showed low overhead video streaming using this technology [23, 24].

Characterizing Communication. The bulk of my work in this area characterized different aspects of communication. A first series of works demonstrated how to measure and characterize high performance commu-

nication interfaces and pipelines [17, 74]. Another paper characterized the communication patterns needed for parallel sorting [19].

I also quantified the relative importance of the parameters of the LogP communication model [16]: latency (L), bandwidth (g), software overhead(o), and computation(P) for a wide variety of parallel programs [51]. I developed a unique slowdown methodology that quantifies the relative importance of individual model parameters. I then showed my methodology was generally applicable by using it to characterize the communication needs of both the Network File System (NFS) as well as the NAS parallel benchmarks [50, 77]. A key contribution of all these works, described in my Ph.D. thesis, is that a very diverse set of applications can tolerate long end-to-end latency, but only combined with low communication software overhead.

In 2001 we quantified how persistent increases in processor speed compared to I/O bus speeds reduce the performance difference between specialized, high performance messaging layers and general purpose protocols such as TCP [35]. Our models show that the performance differential between the two approaches will continue to erode without a radical restructuring of the I/O system. This work completed my involvement in high performance networking.

Related Work. Active Messages demonstrated a general-purpose, high-performance communication layer running over a very diverse set of network interfaces. This result inspired numerous other projects during the explosion of high performance messaging work in 1990's, for example [62, 64], as well related work on network interfaces, such as [7, 12]. Active Messages was unique in its requirement that a message must specify a computation that removes it from the network, although later high performance protocols avoided this constraint [72].

My characterization work is most related to [1, 26]. In both those works, the goal was to extend the LogP model to better describe application characteristics. They are similar to my work in their goal of describing applications in terms of a communication model. My work, however, concentrated on which model parameters are most critical for performance rather than focus on the model itself. I did find that the bandwidth modeling extensions in [1] were necessary to describe many bulk-synchronous programs.

7 Research Philosophy

This final section describes my views on computer systems research. I also relate these themes to the specifics of my research. Finally, I conclude with how my views on these topics translate into my particular research style.

7.1 Defining Computer Systems Research

A common theme throughout my varied research career is its holistic focus on *local computer systems*. Based on my experience with many systems, a limited amount of information available to each component is a unifying characterization across many systems. Thus, my definition of a local computer system is one consisting of interacting computations that have temporal or accessibility limitations with regards to other components' state. Most research challenges arise because components must act without global information needed to make optimal decisions. Examples of such systems include sensor networks, Internet Services, P2P networks, parallel computers, network pipelines, protocol stack layers, and even the operating system. A subtle example is a threaded shared-memory program because typically, one thread does not have access to another's lock acquisition order, and this order is a key part of the thread state. In contrast, a computation that has timely access to every part of itself would not fit this definition. For example, timely access to memory was a property of the PRAM programming model that was seen as a key limitation to developing realistic parallel algorithms [16].

I do not characterize all my research as *distributed systems*, because that term often connotes specific meanings involving virtualization, synchrony, Byzantine behaviors, and the reliability of communication and computation that do not fit many systems. For example, although reasoning about a network pipeline that assumes synchrony, reliable messaging and computation is computer systems work, many researchers would not define such a pipeline as a distributed system. The challenges in managing a pipeline performance arose from the uncertainty in the packet sizes and their arrival and departure times [74].

I find holistic work exploring large systems the most challenging and rewarding, thus all my research fits my definition of local computer systems. In the range-control work, the primary challenge arose because we restricted the algorithm to only use information directly observable by a node; building a perfect topology of the network graph was not an option. Likewise, in the P2P research, many challenges arose because no peer can index all the files in the system or reason when peers enter or leave. We characterized workstation availability using only the observed reboots so that other parts of an Internet Service could reason about likely future behavior without having access to workstation internals. Much of our radio localization research is driven by the low quality of the information concerning other nodes in the system, and then dealing with the resulting uncertainty. Finally, our human factors work can be described in similar terms: the human operator can be cast as a component whose unknown state and resulting unpredictable behavior must be managed by the rest of the system.

There are likely deep reasons for the broad range of fundamental unknowns all these systems face. As [76] points out, even very small programs often generate complex behaviors and are Turing complete. Therefore, their actions are unpredictable and difficult to describe probabilistically even though they are small and deterministic. The components of a computer system are likely to face some similar fundamental uncertainty, which designers must anticipate and work around. However, a key difference between most computer systems and both the classic Byzantine fault tolerance works and chaotic cellular automata is that system components are constructed with known bounds (e.g., the time to restart). Many tricks can be employed to make functional systems using these bounds. Unfortunately, the scope and nature of how all these different systems deal with the lack of information and resulting bounded uncertainty are difficult to generalize.

7.2 Research Style

Computer science is both blessed and cursed because it requires elements of engineering, science, and mathematics. In addition, good research requires an ascetic sense. This qualitative “taste” is critical for picking interesting, tractable research problems that will have an impact in the computer science community. My formal schooling has done much to advance my science, engineering and theory, while my work as an assistant professor has given me much more experience in the ascetics category.

A computer systems research direction must be grounded on a strong theoretical foundation. Ideally, our basic approach for solving a computer systems problem should be based on formalized models that provide confidence in our approach. Often, no such model might exist for the system at hand, so finding or developing a good one becomes a research priority. Indeed, some of the biggest breakthroughs in system design occurred because a formal model balanced simplicity with predictive accuracy under a diverse set of circumstances ¹.

Almost all my work to date has been grounded in formal models. For example, much of the radio-based localization work formalizes uncertainly using Bayesian networks. The density control work used more standard mathematics, and although the resulting equations were only numerically solvable approximations, they turned out to be surprisingly useful in practical settings. The performability work uses very simple algebraic models, but here we chose simplicity over more complex, but less accessible, stochastic models.

Turning to the engineering side, a computer systems researcher should not only be able to design, but more importantly for a researcher, should be able to identify the potential utility resulting from his or her work. That is, if the research were successful, what would be the ultimate impact? Answering these kinds of questions motivates much of the work surrounding computer metrics. Studies measuring performance, availability, performability, and newer human factors metrics are efforts to quantify answers to utility concerns. A second benefit of identifying utility arises because as a human endeavor, computer science research does not exist in a vacuum — it must justify its existence to some human population. Identifying research utility eases this task as well.

I have kept continuous connections to practitioners in the field to address these kinds of utility questions. For example, our recent work surveying database and network administrators is one direct method of gaining information on the practice. Consulting is another, and I have taken leaves and worked as a consultant on several

¹For example, the very simple “iron triangle” model where program execution time = the number of instructions \times the clock rate \times cycles per instruction distilled the interaction of the processor, instruction set, and compiler in a way that greatly advanced all these areas.

occasions. Finally, direct technology transfer, as was the case with my Active Messages work, is another direct validation of the utility of my research.

We need the science component because our artifacts are so large and complex that no one person can predict their behavior based solely on their knowledge of the design. We therefore must turn to the traditional experimental scientific approaches to describe system behavior. Indeed, quantitatively answering the kinds of utility questions described above is impossible without such methods. Nearly all my work to date has thus required designing and performing experiments as well as interpreting the results.

References

- [1] A. Alexandrov, M. Ionescu, K. E. Schauer, and C. Scheiman. LogGP: Incorporating Long Messages into the LogP model - One step closer towards a realistic model for parallel computation. In *7th Annual Symposium on Parallel Algorithms and Architectures*, May 1995.
- [2] S. Asami. Reducing the Cost of System Administration of a Disk Storage System Built from Commodity Components. Technical Report CSD-00-1100, University of California, Berkeley, June 2000.
- [3] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, March 2000.
- [4] R. Barrett, P. P. Maglio, E. Kandogan, and J. Bailey. Usable Autonomic Computing Systems: the Administrator's Perspective. In *Proceedings of the 1st International Conference on Autonomic Computing (ICAC'04)*, May 2004.
- [5] R. Battiti, M. Brunato, and A. Villani. Statistical Learning Theory for Location Fingerprinting in Wireless LANs. Technical Report DIT-02-086, University of Trento, Informatica e Telecomunicazioni, Oct. 2002.
- [6] R. Bianchini, R. P. Martin, K. Nagaraja, T. D. Nguyen, and F. Oliveira. Human-aware computer system design. In *The 10th Workshop on Hot Topics in Operating Systems (HotOS)*, June 2005.
- [7] M. A. Blumrich, K. Li, R. Alpert, C. Dubnicki, E. Felten, and J. Sandberg. Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer. In *Proceedings of the 21st International Symposium on Computer Architecture*, Apr. 1994.
- [8] P. M. Broadwell. Response Time as a Performability Metric for Online Services. Technical Report UCB//CSD-04-1324, University of California At Berkeley, Computer Science Division, May 2003.
- [9] A. Brown. Towards availability and maintainability benchmarks: a case study of software RAID systems. Master's thesis, Computer Science Division-University of California, Berkeley, Dec. 2001.
- [10] A. B. Brown and D. A. Patterson. Undo for Operators: Building an Undoable E-mail Store. In *Proceedings of the 2003 USENIX Annual Technical Conference*, June 2003.
- [11] Y. Chen, W. Trappe, , and R. P. Martin. Robustness of Localization Algorithms to Attacks: A Comparative Study. In Preparation, 2005.
- [12] D. Chiou, B. Ang, Arvind, M. Beckerle, G. Boughton, R. Greiner, J. Hicks, and J. Hoe. StarT-NG: Delivering Seamless Parallel Computing. In *EURO-PAR'95 Conference*, Aug. 1995.
- [13] A. Choksi, R. P. Martin, B. Nath, and R. Pupala. Mobility Support for Diffusion-based Ad-Hoc Sensor Networks. Technical Report DCS-463, Rutgers University Department of Computer Science, apr 2002.
- [14] F. M. Cuenca-Acuna, R. P. Martin, and T. D. Nguyen. Autonomous replication for high availability in unstructured P2P systems. In *Proceedings of the IEEE 23rd International Symposium on Reliable Distributed Systems SRDS*, Oct. 2004.

- [15] F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In *Proceedings of the IEEE International Symposium on High Performance Distributed Computing (HPDC)*, June 2003.
- [16] D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauer, E. Santos, R. Subramonian, and T. von Eicken. LogP: Towards a Realistic Model of Parallel Computation. In *Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 262–273, 1993.
- [17] D. E. Culler, L. T. Liu, R. P. Martin, and C. O. Yoshikawa. Assessing Fast Network Interfaces. In *IEEE Micro*, volume 16, pages 35–43, Feb. 1996.
- [18] L. Doherty¹, K. S. J. Pister, and L. E. Ghaoui. Convex Position Estimation in Wireless Sensor Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, Apr. 2001.
- [19] A. C. Dusseau, D. E. Culler, K. E. Schauer, and R. P. Martin. Fast Parallel Sorting Under LogP: Experience with the CM-5. In *IEEE Transactions on Parallel and Distributed Systems*, volume 7, pages 791–805, 1996.
- [20] E. Elnahrawy, J.-A. Francisco, and R. P. Martin. Bayesian Localization in Wireless Networks using Angle of Arrival. Under Review, July 2005.
- [21] E. Elnahrawy, X. Li, and R. P. Martin. The limits of Localization Using Signal Strength: A Comparative Study. In *In Proceedings of the IEEE Conference on Sensor and Ad Hoc Communication Networks (SECON)*, Oct. 2004.
- [22] E. Elnahrawy, X. Li, and R. P. Martin. Using Area-based Presentations and Metrics for Localization Systems in Wireless LANs. In *4th International IEEE Workshop on Wireless Local Networks (WLN 2004)*, Tampa, FL, Nov. 2004.
- [23] M. E. Fiuczynski, R. P. Martin, T. Owa, and B. N. Bershad. On using intelligent network interface cards to support multimedia applications. In *The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, July 1998.
- [24] M. E. Fiuczynski, R. P. Martin, T. Owa, and B. N. Bershad. SPINE: a safe programmable and integrated network environment. In *ACM SIGOPS European Workshop*, Sept. 1998.
- [25] I. Foster and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.
- [26] M. I. Frank, A. Agarwal, and M. Vernon. LoPC: Modeling Contention in Parallel Algorithms. In *Proceedings of SIXTH ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, June 1997.
- [27] C. Fu, R. P. Martin, K. Nagaraja, T. D. Nguyen, B. G. Ryder, and D. Wonnacott. Compiler-directed program-fault coverage for highly available java internet services. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, 2003.
- [28] G. M. C. Gama, K. Nagaraja, R. Bianchini, R. P. Martin, W. Meira Jr., and T. D. Nguyen. State maintenance and its impact on the performability of multi-tiered internet services. In *Proceedings of the IEEE 23rd International Symposium on Reliable Distributed Systems SRDS*, Oct. 2004.
- [29] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi. Analysis of Preventive Maintenance in Transactions Based Software Systems. *IEEE Transactions on Computers*, 47(1):96–107, Jan. 1998.
- [30] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2001.

- [31] J. Gray. Why do Computers Stop and What Can Be Done About It? In *Proceedings of 5th Symposium on Reliability in Distributed Software and Database Systems*, Jan. 1986.
- [32] J. Gray. A Census of Tandem System Availability Between 1985 and 1990. *IEEE Transactions on Reliability*, 39(4):409–418, Oct. 1990.
- [33] M. Hazas and A. Ward. A high performance privacy-oriented location system. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Dallas, TX, Mar. 2003.
- [34] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. In *Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, Sept. 2003.
- [35] T. Heath, S. Kaur, R. P. Martin, and T. D. Nguyen. Quantifying the impact of architectural scaling on communication. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, Jan. 2001.
- [36] T. Heath, R. Martin, and T. D. Nguyen. The Shape of Failure. In *Proceedings of the First Workshop on Evaluating and Architecting System dependability (EASY)*, July 2001.
- [37] T. Heath, R. Martin, and T. D. Nguyen. Improving Cluster Availability Using Workstation Validation. In *Proceedings of the ACM SIGMETRICS 2002*, Marina Del Rey, CA, June 2002.
- [38] P. J. K. Jr., J. Sung, C. P. Dingman, D. P. Siewiorek, and T. Marz. Comparing Operating Systems Using Robustness Benchmarks. In *Proceedings of the Symposium on Reliable Distributed Systems (SRDS'97)*, pages 72–79, 1997.
- [39] M. Kalyan Krishnam, Z. Kalbarczyk, and R. Iyer. Failure Data Analysis of a LAN of Windows NT Based Computers. In *Proceedings of the 18th Symposium on Reliable and Distributed Systems (SRDS '99)*, 1999.
- [40] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1), Jan. 2003.
- [41] I. Lee and R. K. Iyer. Faults, Symptoms, and Software Fault Tolerance in the Tandem GUARDIAN90 Operating System. In *Proceedings of International Symposium on Fault-Tolerant Computing (FTCS-23)*, pages 20–29, 1993.
- [42] X. Li and R. P. Martin. A Simple Ray-Sector Signal Strength Model for Indoor 802.11 Networks. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, Nov. 2005.
- [43] X. Li, R. P. Martin, K. Nagaraja, T. D. Nguyen, and B. Zhang. Mendosus: SAN-Based Fault-Injection Test-Bed for the Construction of Highly Available Network Services. In *Proceedings of 1st Workshop on Novel Uses of System Area Networks (SAN-1)*, Jan. 2002.
- [44] X. Li, T. D. Nguyen, and R. P. Martin. An Analytic Model Predicting the Optimal Range for Maximizing 1-Hop Broadcast Coverage in Dense Wireless Networks. July 2004.
- [45] X. Li, T. D. Nguyen, and R. P. Martin. Using Adaptive Range Control to Maximize 1 Hop Broadcast Coverage in Dense Wireless Networks. In *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communication Networks (SECON)*, Oct. 2004.
- [46] T. Liu, Z. Kalbarczyk, and R. Iyer. A Software, Multilevel Fault Injection Mechanism: Case Study Evaluating the Virtual Interface Architecture. In *Proceedings of the Symposium on Reliable Distributed Systems (SRDS'99)*, Lausanne, Switzerland, 1999.

- [47] D. D. E. Long, J. L. Carroll, and C. J. Park. A Study of the Reliability of Internet Sites. In *Proceedings of the Tenth Symposium on Reliable Distributed Systems*, pages 177–186, Sept. 1991.
- [48] D. Madigan, E. Elnahrawy, R. P. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar. Bayesian Indoor Positioning Systems. In *Infocom*, 2005.
- [49] R. P. Martin. HPAM: An Active Message Layer for a Network of Workstations. In *Proceedings of the 2nd Hot Interconnects Conference*, July 1994.
- [50] R. P. Martin and D. E. Culler. NFS Sensitivity to High Performance Networks. In *ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, May 1999.
- [51] R. P. Martin, A. M. Vahdat, D. E. Culler, and T. P. Anderson. The Effects of Latency, Overhead and Bandwidth in a Cluster of Workstations. In *Proceedings of the 24th International Symposium on Computer Architecture*, Denver, CO, June 1997.
- [52] J. F. Meyer. Performability evaluation: Where it is and what lies ahead. In *Proceedings of the IEEE International Computer Performance and Dependability Symposium*, pages 334–343, Erlangen, Germany, Apr. 1995.
- [53] B. Murphy and B. Levidow. Windows 2000 Dependability. Technical Report MSR-TR-2000-56, Microsoft Research, June 2000.
- [54] K. Nagaraja, R. Bianchini, R. Martin, and T. D. Nguyen. Using Fault Model Enforcement to Improve Availability. In *Proceedings of the 2nd Workshop on Evaluating and Architecting System dependability (EASY)*, Oct. 2002.
- [55] K. Nagaraja, G. M. C. Gama, R. Bianchini, R. P. Martin, W. Meira Jr., and T. D. Nguyen. Quantifying the performability of cluster-based services. *IEEE Transactions on Parallel Distributed Systems*, 16(5):456–467, 2005.
- [56] K. Nagaraja, N. Krishnan, R. Bianchini, R. Martin, and T. D. Nguyen. Evaluating the Impact of Communication Architecture on the Performability of Cluster-Based Services. In *Proceedings of the 9th Symposium on High Performance Computer Architecture (HPCA-9)*, Feb. 2003.
- [57] K. Nagaraja, X. Li, B. Zhang, R. Bianchini, R. P. Martin, and T. D. Nguyen. Using Fault Injection and Modeling to Evaluate the Performability of Cluster-Based Services. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS'03)*, Mar. 2003.
- [58] K. Nagaraja, F. Oliveira, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and Dealing with Operator Mistakes in Internet Services. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI '04)*, Dec. 2004.
- [59] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS). In *GLOBECOM (1)*, pages 2926–2931, 2001.
- [60] F. Oliveira, R. Bachwani, K. Nagaraja, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and validating database system administration. In preparation, 2005.
- [61] D. Oppenheimer, A. Ganapathi, and D. Patterson. Why do Internet Services Fail, and What Can Be Done About It. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS'03)*, Mar. 2003.
- [62] S. Pakin, M. Lauria, and A. Chien. High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet. In *Supercomputing '95*, San Diego, California, 1995.

- [63] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support system. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000.
- [64] S. H. Rodrigues, T. E. Anderson, and D. E. Culler. High-Performance Local-Area Communication Using Fast Sockets. In *Proceedings of the 1997 USENIX Annual Technical Conference*, Anaheim, CA, Jan. 1997.
- [65] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Nov. 2001.
- [66] A. Savvides, C.-C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Proceedings of the Seventh Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.
- [67] R. M. Smith, K. S. Trivedi, and A. V. Ramesh. Performability Analysis: Measures, an Algorithm, and a Case Study. *IEEE Transactions on Computers*, 37(4), April 1998.
- [68] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Aug. 2001.
- [69] M. Sullivan and R. Chillarege. Software Defects and their Impact on System Availability - A Study of Field Failures in Operating Systems. In *Proceedings of the 21st International Symposium on Fault-Tolerant Computing (FTCS-21)*, pages 2–9, Montreal, Canada, 1991.
- [70] A. Tjang, M. Pagliorola, H. Patel, X. Li, and R. P. Martin. Active tapes: Bus-based sensor networks. Technical Report DCS-TR-557, Rutgers University Department of Computer Science, June 2004.
- [71] T. Tsai, M. Hsueh, H. Zhao, Z. Kalbarczyk, and R. Iyer. Stress-based and path-based fault injection. *IEEE Transactions on Computers*, 48(11):1183–1201, Nov. 1999.
- [72] T. von Eicken, A. Basu, V. Buch, and W. Vogels. U-Net: A User-Level Network Interface for Parallel and Distributed Computing. In *Proceedings of the Fifteenth SOSP*, pages 40–53, Copper Mountain, CO, December 1995.
- [73] T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauer. Active Messages: a Mechanism for Integrated Communication and Computation. In *Proc. of the 19th Int'l Symposium on Computer Architecture*, May 1992.
- [74] R. Wang, A. Krishnamurthy, R. P. Martin, T. Anderson, and D. E. Culler. Modeling and Optimizing Communication Pipelines. In *Proceedings of the 1998 ACM SIGMETRICS and PERFORMANCE Conference on Measurement and Modeling of Computer Systems*, Madison, WI, June 1998.
- [75] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, Jan. 1992.
- [76] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.
- [77] F. C. Wong, R. P. Martin, R. H. Arpaci-Dusseau, and D. E. Culler. Architectural requirements and scalability of the NAS parallel benchmarks. In *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, Nov. 1999.
- [78] M. Youssef, A. Agrawal, and A. U. Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of IEEE PerCom'03*, Fort Worth, TX, Mar. 2003.
- [79] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2), Mar. 2002.

- [80] Y. Zhao, J. Kubiawicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, University of California, Berkeley, April 2000.