David G. Andersen

MIT Lab for Computer Science
NE43-512
200 Technology Square
Cambridge, MA 02139
(617) 452-2820
`dga@lcs.mit.edu`
`http://www.angio.net/personal/`

Education

MASSACHUSETTS INSTITUTE OF TECHNOLOGY — Cambridge, MA

S.M. in Computer Science, 2001
Ph.D. candidate in Computer Science. (Expected summer 2004)
Advisor: Hari Balakrishnan
Minor in Computational Biology

UNIVERSITY OF UTAH — Salt Lake City, UT

Bachelor of Science in Computer Science. *Cum Laude*, 1998
Bachelor of Science in Biology. *Cum Laude*, 1998

Research Interests

Computer systems in the networked environment.

Professional Experience

1999– **Research Assistant** — MIT

Research assistant at the Laboratory for Computer Science (LCS / CSAIL). Worked in cooperation with the University of Utah on the RON+Emulab testbed. Major projects at MIT include Resilient Overlay Networks, Resilient Access Networks, Mayday, and the Congestion Manager. A summary of my research activities at MIT and elsewhere begins on page 4.

Summer 2001 **Summer Intern** — Compaq SRC

Summer internship working on the Secure Network Attached Disks project.

1997-1999 **Research Assistant / Research Associate** — University of Utah

One year as an undergraduate and one year as a staff research associate in the Flux research group at the University of Utah.

1996-1997 **Research Assistant** — Department of Biology, University of Utah

Undergraduate research assistantship in the Wayne Potts Laboratory in the Department of Biology.

1995-1997 **Co-founder and CTO, ArosNet, Inc.**

Acted in a directorial and technical capacity over technical operations: network design and topology planning, software development, consulting projects, and short-term research. During my three years with the company, ArosNet grew from its inception to become the third largest ISP in Utah.

1995-2003 **Consultant** — *IJNT, Inc., Sypherance Technologies, Ascensus, others*.

Provided network design, security, and intellectual property consulting services.

1993-1995 **Systems Administrator, The Lower Lights**

Implemented and managed database systems and medium-scale dialin analog modem banks.

Teaching Experience

2001 — Teaching Assistant, MIT course 6.829, Computer Networks.
Assisted with homework / quiz design and grading, some lectures, office hours. Designed and coordinated the group project for undergraduate students.

2001-2003 — UROP / Undergraduate Thesis supervisor, MIT.
Co-supervised one undergraduate thesis and two undergraduate research assistants in the Congestion Manager and RON projects.

1997 — Teaching Assistant, University of Utah course CS508, Computer Networks.
Assisted with Homework / quiz design and grading, held weekly office hours.

2001-2003 — Lecturer and organizer, MIT Winter Mountaineering class.
Organized curriculum and lectures, presented some lectures.

Refereed Publications

[1] David G. Andersen, Alex C. Snoeren, and Hari Balakrishnan. Best-path vs. multi-path overlay routing. In *Proc. Internet Measurement Conference*, October 2003.

[2] Nick Feamster, David Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *Proc. ACM SIGMETRICS*, San Diego, CA, June 2003.

[3] Marcos K. Aguilera, Minwen Ji, Mark Lillibridge, John MacCormick, Erwin Oertli, David G. Andersen, Mike Burrows, Timothy Mann, and Chandramohan Thekkath. Block-Level Security for Network-Attached Disks. In *Proc. 2nd USENIX Conference on File and Storage Technologies (FAST)*, March 2003.

[4] David G. Andersen. Mayday: Distributed Filtering for Internet Services. In *Proc. USENIX Symposium on Internet Technologies and Systems (USITS)*, March 2003.

[5] David G. Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan. Topology Inference from BGP Routing Dynamics. In *Proc. Internet Measurement Workshop*, Marseille, France, November 2002.

[6] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proc. 18th ACM SOSP*, pages 131–145, Banff, Canada, October 2001.

[7] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. The Case for Resilient Overlay Networks. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HOTOS-VIII) (Best Student Paper Award)*, May 2001.

[8] Alex Snoeren, David Andersen, and Hari Balakrishnan. Fine-Grained Failover Using Connection Migration. In *Proc. USENIX Symposium on Internet Technologies and Systems (USITS)*, September 2001.

[9] David Andersen, Deepak Bansal, Dorothy Curtis, Srinivasan Seshan, and Hari Balakrishnan. System Support for Bandwidth Management and Content Adaptation in Internet Applications. In *Proc. of the Fourth Symposium on Operating Systems Design and Implementation*, October 2000.

[10] Ray Spencer, Stephen Smalley, Peter Loscocco, Mike Hibler, David Andersen, and Jay Lepreau. The Flask Security Architecture: System Support for Diverse Security Policies. In *Proc. of the Eighth USENIX Security Symposium*, August 1999.

Other Articles

[11] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Experience with an Evolving Overlay Network Testbed. *Computer Communication Review*, 33(3):13–19, July 2003.

Pending and Submitted Publications

[12] David G. Andersen, Hari Balakrishnan, and Frans Kaashoek. Grassroots Reliability with Resilient Access Networks. To be submitted, February, 2004.

[13] Jeffrey Considine, Michael Walfish, and David G. Andersen. A pragmatic approach to DHT adoption. Submitted for publication, November, 2003.

Patents

"Method and system for securing block-based storage with capability data." Marcos K. Aguilera, Minwen Ji, Mark Lillibridge, John MacCormick, Oerwin Oertli, Dave Andersen, Mike Burrows, Tim Mann, Chandu Thekkath. Pending, filed in May 2003.

Selected Honors and Awards

2002–2004	Microsoft Research Graduate Fellowship
2001	Best Student Paper, 8th IEEE Workshop on Hot Topics in Operating Systems
2001	MIT Joseph Levin award for best MasterWorks oral presentation
1999	MIT Vinton Hayes Fellowship (graduate)
1998	University of Utah Graduating Student Leadership Award
1993	Member, Phi Kappa Phi and Golden Key academic honor societies
1993–1997	University of Utah Honors at Entrance Scholarship
1993	National Merit Scholar

Service and Other Activities

	Reviewer for OSDI, SOSP, SIGCOMM, CCR, HotOS, ToN, Infocom, HotNets.
1999–2003	Secretary, board member, and rock climbing instructor for the MIT Outing Club.
1999–2000	Secretary, Utah Regional Exchange Point
1997–1998	Chair, University of Utah Undergraduate CS Advisory Committe
1995-1996	EMT Volunteer, University of Utah Medical Center emergency room.

Research - Network Resilience and Performance

1999– **Resilient Overlay Networks** MIT

My dissertation research investigates host-based techniques that improve the end-to-end fault resilience of communication on the Internet. Wide-area reachability suffers two weaknesses. First, inter-provider routing with BGP can be fragile and suffers from a longer time-to-repair than does intra-provider routing. Second, clients' access links are a common single point of failure impacting end-to-end reachability. The Resilient Overlay Networks (RON) and Resilient Access Networks (RAN) projects address these two points of failure.

RON is a framework that creates dynamic overlay networks between participating hosts or applications. The overlay networks use a combination of active probing and passive measurements to find more reliable and better performing routes by sending packets through the other participating nodes in the overlay. A set of Internet-based experiments in 2001 showed that RON can avoid up to half of the failures that interrupt communication, and can offer significant latency improvements for poorly-performing paths.

RAN uses a combination of overlay techniques and multiple local Internet connections to improve clients' connectivity to Internet hosts. RAN improves the reliability of hosts' communication not only with each other, but also to external hosts, and incorporates techniques that address some of the scalability concerns of RON. The RAN system is currently implemented in a Web proxy system running and being evaluated at MIT, the University of Utah, and two private companies.

1999–2000 **Congestion Manager** MIT

The Congestion Manager provides a unified congestion controller for ensembles of TCP and UDP flows that eliminates adverse interactions and extends the benefits of congestion control to non-TCP applications. To help evaluate the CM, I co-implemented a congestion-controlled version of `vat`, an internet audio tool, which used the Congestion Manager to behave in a TCP-friendly manner with low overhead. I helped design and implement the kernel to user API for the CM, and performed extensive performance measurements of the CM for both in-kernel and userspace applications.

1998– **Emulab + RON Testbed** University of Utah / MIT

Systems and networking researchers frequently use home-grown testbeds to evaluate prototypes and perform Internet measurements. To reduce the burden of creating these testbeds and to help provide a framework with better experimental repeatability, I played a part in the conception and design of a large-scale network testbed, Emulab, and a portion of its management databases, algorithms, and software. At MIT, I deployed and currently manage a 36-node distributed Internet testbed which is integrated with Emulab. I have been helping to transition the lessons learned from this testbed into the emerging Planetlab testbed.

Research - Network Security

2003 **Mayday: Distributed Filtering for Internet Services** MIT

Mayday presents an incrementally deployable Denial of Service *prevention* service that acts primarily as an overlay service, minimizing the network changes required for its deployment. Unlike tactics such as spoofing prevention, Mayday provides immediate protection to its deployers instead of requiring upgrades on the part of third parties. Mayday generalizes earlier work on Secure Overlay Services by separating overlay routing from filtering and by providing a larger set of choices for each, allowing the implementer to choose a high-performance deployment such as proximity routing, or a slower system that can withstand more capable attackers.

As part of the evaluation of Mayday and earlier work, I developed several practical attacks, two of them novel, that are effective against filtering-based systems like Mayday and SOS.

Summer 2001 **Secure Network Attached Disks** Compaq SRC

Traditional disk architectures interpose a fileserver between clients and disks to provide access control. *Network Attached Disk* efforts aim to place the disks directly on the network, eliminating the bottleneck presented by the file server. The capability-based approach we examined permits the disks to export a familiar block-based interface; compared to earlier NAD efforts, this eliminates disk layout changes and simplifies the on-disk implementation. I created a filesystem simulator for our proposed architecture and created a benchmark suite from measurements of SRC's fileserver traffic to drive the simulator.

1997-1999 **Flask: A secure microkernel** University of Utah

Users' requirements for operating systems vary considerably, from the MLS policies favored in military applications, to RBAC-like policies more common in large enterprises, to type enforcement policies favored for providing least privilege to local processes. The Flask security architecture provides fine-grained access rights and permits for their revocation to permit a single OS implementation to support a wide range of security policies. As an undergraduate, and continuing as research staff, I implemented and benchmarked parts of the Flask architecture, improved the reliability of the underlying Fluke microkernel, and implemented several of the example applications used in its evaluation.

References

Prof. Hari Balakrishnan
MIT Laboratory for Computer Science
200 Technology Square, NE43-510
Cambridge, MA 02139
(617) 253-8713
hari@lcs.mit.edu

Prof. M. Frans Kaashoek
MIT Laboratory for Computer Science
200 Technology Square, NE43-522
Cambridge, MA 02139
(617) 253-7149
kaashoek@lcs.mit.edu

Prof. John Guttag
MIT Dept. Electrical Engineering & Computer Science
38-401
77 Massachusetts Avenue
Cambridge, MA 02139
(617) 253-6022
guttag@mit.edu

Prof. Larry L. Peterson
Princeton University
Department of Computer Science
35 Olden St.
Princeton, NJ 08544
(609) 258-6077
llp@cs.princeton.edu

Prof. Jay Lepreau
University of Utah School of Computing
50 S. Central Campus Dr. Rm 3190
Salt Lake City, UT 84112
(801) 581-4285
lepreau@cs.utah.edu

Research Statement for David Andersen

I enjoy conducting networking and systems research in a way that combines tactics from my backgrounds in both computing and the experimental sciences. In my research, I seek to create useful software systems, deploy them in real-world settings, and perform extensive experiments to characterize and explain their behavior and performance. Because modern computer systems operate in an ecosystem (such as the Internet), not a vacuum, experimental data obtained in the process of measuring deployed systems reveals important information about the underlying environment as well.

The ultimate goal of systems research, however, is the production of better systems. To this end, creating working solutions to real problems provides both great incentive and a necessary sanity check on the results of systems research. Much of my future work will retain this dual theme of experimentation and systems-building.

A major focus throughout my research has been on implementing and evaluating systems that perform well during failures and outages. Individual components of the Internet are well-engineered and often highly reliable, but the global ensemble of users, ISPs, access links, and software fails more often than we would like. I have worked on two distributed systems, RON (Resilient Overlay Networks) [3] and RAN (Resilient Access Networks) [2], which provide end-to-end mechanisms to discover and recover from Internet faults on fast time scales. RON operates by constantly measuring the paths between a small set of cooperating computers and sending data indirectly via a third computer if the direct path is unusable. RAN sends multiple session initiation packets and uses them as probes to determine a working path. In this manner, RAN provides resiliency to legacy protocols like HTTP and DNS.

Both RON and RAN have been deployed in real settings, allowing me to conduct numerous experiments. My results show that the simple approaches work well at overcoming outages: In a 20-node Internet experiment, RON avoided up to 50% of the outages between communicating hosts. In its 8 months of deployment as a Web proxy in MIT's CSAIL, RAN has successfully masked about half of the incidents in which servers were unreachable, and eliminated 75% of certain pathological delays. The original RON research opened up considerable discussion about its scalability and fairness, which inspired many follow-on research projects, including my own work on RAN. In addition, at least one commercial product uses RON's techniques to provide highly reliable distributed storage.

These reliability challenges extend beyond the Internet: complex software systems and locally distributed systems often present failures that are dauntingly hard to debug and prevent. Like systemically over-provisioned networks, today's computer systems often have excess power and storage. RON and RAN provide a way to use excess network capacity to improve resiliency, and I believe that a similar approach may work for other systems. I would like to examine systems from the perspective of reliability and dependability, applying conceptually simple (but perhaps more resource-intensive) techniques that can greatly improve their reliability. Approaches that improve service availability may apply in these systems to deal with component, as well as network, failures, creating a more unified outage masking architecture.

One contribution of the RON project is that it treated the overlay network as a first-class network, not just as a vehicle for content distribution. This insight contributed similarly to my work on Mayday [1], in which the separation of overlay routing and network filtering improves both the efficiency and security of denial-of-service (DoS) prevention techniques. By examining these techniques in a realistic environment, Mayday also presented several practical and effective techniques against which future filtering schemes should be robust. The insights from Mayday about the strengths *and* weaknesses of overlay-based systems contributed to several on-going research efforts towards denial of service prevention techniques.

I believe that a thorough understanding of methods for securing computer systems and applying "security thinking" from the ground up is critical in the design and implementation of modern software systems. Building systems for real-world deployment results in an inevitable confrontation with security challenges. In the future, I plan further investigation of systems that, like Mayday, explore the boundaries and interactions between application security and network security. My experience with these systems suggests that there is a considerable efficiency trade-off involved in using only end-system techniques to achieve network security, but these techniques are typically easier to deploy and provide a good incentive structure for deployment. I don't know how far an end-system only approach can be pushed to improve network availability, but I believe that its deployment advantages make it an attractive subject for research.

Developing and sharing tools and techniques that facilitate further research is, I believe, an important duty of systems research. As part of the RON research, I deployed a medium-size Internet testbed of 36 nodes spread over 8 countries. The testbed proved essential to measuring the effectiveness of the RON approach, but was also a useful research artifact on its own—at least twelve other research projects used the testbed to gather data or evaluate their own systems, and it is currently in use by additional projects [4]. I've had the chance to contribute some of the knowledge gained from the RON testbed to the emerging Planetlab project, and have been involved since its planning stages with the design and implementation of the Emulab testbed. I believe that creating these testbeds and making them available to the wider research community can substantially improve the state of systems research, and I intend to work further on all three projects—preferably together.

Accompanying the RON testbed is a large centralized database that collects Internet measurements and routing updates from the testbed nodes to facilitate later analysis. With my colleagues, I've used the database to analyze the correlation between Internet outages and BGP updates [6], finding that most outages precede BGP messages by about four minutes, but that 20% of the failures could actually be *predicted* by a string of BGP messages. This study wasn't possible without data that combined both end-to-end probing data *and* backbone routing traces.

I also enjoy bringing new tools to bear on systems and networks problems. Using the BGP database, I showed that statistical clustering methods can be used on BGP traces to infer the internal topology of a remote network, even if that network blocks traceroute and other probes [5]. These clustering techniques are widely used in other fields for data exploration (I encountered them in the context of computational genomics); that they applied so well to network data suggests great potential for the application of other data-mining techniques to Internet measurement data.

These studies were feasible only *once the data was available.* A constant problem facing networking researchers is the lack of coherent, real-world data about network performance, and I believe we have barely scratched the surface of the utility of a well-organized, easily accessible repository of traffic, routing, and measurement data. One of my major research goals is to explore new, scalable architectures that facilitate the collection and aggregation of these kinds of data and make them available in an easily usable format to other researchers and to systems that can make use of the data in real-time. An approach that combines archival use with active use offers the potential of reduced overhead and duplication of effort, and the possibility of creating a long-term, high quality repository of measurements researchers desperately need.

I plan to continue exploring issues of resilient networked systems in the future. Resource-poor environments—lacking computation, bandwidth, connectivity, and financial resources—present an increased set of challenges for reliable systems. The growing popularity of both fixed and mobile wireless systems means that these challenges will soon be ever more present in deployed systems.

While RON and RAN worked well, I believe that broadening their goal somewhat can yield many more benefits. RON and RAN improved connectivity between specific computers, or between a computer and a service. In some cases, such as real-time communication, the RON approach matches the users needs. In many cases, however, users do not care about a connection to a computer, they care about access to *information*: an email message, a web page, their medical records, etc.. For these examples, a more powerful set of techniques can be used to improve users' ability to access them. Background replication, prefetching, alternate or redundant transport mechanisms, and other techniques can all be brought to bear. While all of these techniques could be used at the application layer, the third part of my future research will be exploring a middle ground that applies to a wide range of systems while providing most of the benefits of application integration. I don't know the exact form of this middle ground (streams? objects? opaque data blocks?) but I believe we can create an effective common layer for building much more resilient systems.

References

[1] David G. Andersen. Mayday: Distributed Filtering for Internet Services. In *Proc. USENIX Symposium on Internet Technologies and Systems (USITS)*, March 2003.

[2] David G. Andersen, Hari Balakrishnan, and Frans Kaashoek. Grassroots Reliability with Resilient Access Networks. To be submitted, February, 2004.

[3] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proc. 18th ACM SOSP*, pages 131–145, Banff, Canada, October 2001.

[4] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Experience with an Evolving Overlay Network Testbed. *Computer Communication Review*, 33(3):13–19, July 2003.

[5] David G. Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan. Topology Inference from BGP Routing Dynamics. In *Proc. Internet Measurement Workshop*, Marseille, France, November 2002.

[6] Nick Feamster, David Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *Proc. ACM SIGMETRICS*, San Diego, CA, June 2003.

Education Statement for David Andersen

I believe that the most important job a teacher can do is to inspire in students interest and passion about the material. Excited students will learn more about a subject than the teacher ever imagined, and disinterested students will leave their knowledge at the door. My own interest in teaching, research, and particularly in operating systems was strongly influenced by a great undergraduate operating systems course. While taking that course, I really started to see how amazing computing is, with its combination of elegant system designs, algorithmic design, and the joy (and suffering!) of creating real, working systems. I want to engender that kind of enthusiasm in my own students.

I was fortunate to be allowed to help TA the networks course at the University of Utah while an undergraduate there. At MIT, I was the head TA for the graduate networks class (6.829). One of the most gratifying aspects of this course was working with the students on the small research projects they undertook for their course projects. Students had the opportunity to pick their own research project or to implement a project that I designed in which they built a multi-site file distribution system.

In my experience, there are two ways to help students get excited about a course or a project. The first is to make the relevance of the material clear. For instance, many pertinent systems topics, such as routing, caching, and consistency are explainable via modern peer-to-peer systems with which students have personal experience. The second is by making the course hands-on through experiments and projects. These techniques work best in “lab” courses, which are the courses I’m most interested in teaching: advanced networking and computer systems classes at the graduate level, and networking, OS, and introductory programming courses at the undergraduate level.

Systems classes need to blend the principles underlying computing systems with enough of the implementation details and hands-on expereience to give students an intuition for the concepts; the experience necessary to apply the principles; and a reality-based lens through which they can critically examine both solutions to future problems *and* the course material. Real-world problems involve both the application of theory (e.g., wireless MAC protocols, synchronization techniques, algorithms, etc.) and experience with the problems that arise when using, designing, implementing, and debugging real systems. I believe that classes with a strong hands-on component are the best way to expose students to both aspects of systems.

For most students, a good computer science education occurs as much outside the classroom as it does inside. Whether this happens through involvement in research, independent projects, or interaction with industry, these outside activities and larger course projects introduce students to open-ended creative problems in which they are responsible for both the design and the challenges of implementing their design. I believe that it’s important to give students the opportunity, at both the graduate and undergraduate level, to be involved in research and real system building.

To this end, systems classes offer students a springboard to further involvement with a faculty member’s research, an experience I believe is valuable both for research and industry-bound students. In the MIT networks course I TA’d, several groups of students went on to turn their course projects into full-fledged research papers, something I hope to continue in my own courses and in my research group. From that networks course, I also met the three undergraduate students whom I mentored on their UROP (undergraduate research opportunity) and senior thesis projects. I learned as much during this mentoring as my students did, and I would like to have my own graduate students participate in the co-supervision of undergraduate projects.

In addition to computer science teaching, I also organized, helped design, and lectured for the MIT Outing Club's month-long winter mountaineering classes. Having the opportunity to design my own course from the ground up was as gratifying as it was time intensive, and the experience confirmed for me that I truly enjoy both the process of helping people learn and the amazing education that one gains while teaching others.