Michael J. Freedman

353 Serra Mall, Room 288
Stanford, CA 94305-9025
(650) 723-1863

http://www.michaelfreedman.org/
mfreed@scs.stanford.edu
Citizenship: US

Education

New York University .... New York, NY

Ph.D. Candidate in Computer Science .... Expected May 2007
Thesis title: *Harnessing Widespread Cooperation to Democratize Content Distribution*
Visiting **Stanford University**, September 2005–May 2007
Advisor: David Mazières ; GPA: 4.0/4.0

M.S. in Computer Science .... May 2005
Advisor: David Mazières ; GPA: 4.0/4.0

Massachusetts Institute of Technology .... Cambridge, MA

M.Eng. in Electrical Engineering and Computer Science .... June 2002
Thesis title: *A Peer-to-Peer Anonymizing Network Layer*
Advisor: Robert Morris ; GPA: 5.0/5.0

S.B. in Computer Science, Minor in Political Science .... June 2001
Thesis title: *An Anonymous Communications Channel for the Free Haven Project*
Advisor: Ron Rivest ; GPA: 4.9/5.0

Interests

Distributed systems, security, networking, and cryptography

Research

2002–present **Cooperative content distribution.** Conceived and led the Coral Project. Designed and built an Internet-scale, self-organizing web-content distribution network: CoralCDN [11] uses a network of cooperating DNS redirectors and HTTP proxies, backed by a decentralized indexing infrastructure [18], to allow oblivious clients to transparently download content from nearby servers, while avoiding distant or heavily-loaded ones. CoralCDN has been in production use on 300 servers since March 2004, currently receiving about 25 million HTTP requests from over 1 million clients per day, serving several terabytes of data. `http://coralcdn.org/`

With a focus on settings with mutually-distrustful clients, Shark [6] provides a distributed file system that improves scalability and performance through cooperative reads, using Coral's indexing layer to locate files. Yet Shark preserves traditional semantics, manageability, and security. Other research provides integrity guarantees for large files encoded with rateless erasure codes, via a homomorphic hash function that can verify downloaded blocks on-the-fly [10].

Ongoing focus on untrusted settings for CDNs (with C. Aperjis, R. Johari, and D. Mazières), devising incentive-compatible mechanisms that cause nodes to contribute bandwidth for improved quality-of-service. This work uses market-pricing techniques and virtual currency to ensure effective bandwidth usage and network utilization, while still preventing cheating.

2005–present **Anycast.** Designed and built OASIS, a server-selection infrastructure that provides locality- and load-based anycast for replicated Internet services [3] [25]. OASIS tackles the problems of leveraging disparate services to perform (potentially error-prone) network measurement and of scalably managing state information about many services and their participating nodes. OASIS has been in production use since Nov. 2005 and has been adopted by more than a dozen distributed services, handling thousands of replicas. Performed background studies of the geographic locality of IP prefixes [5] and the efficacy of virtual coordinate systems [16]. `http://oasis.coralcdn.org/`

2006–present **IP analytics.** By instrumenting CoralCDN, used active web content to measure and analyze the characteristics of over 7 million clients with respect to "edge technologies" (NATs, proxies, DNS

	and DHCP) [1]. Results quantify how Internet services can use IP addresses to identify clients and enforce access-control decisions. Commercialized historical and real-time techniques for proxy detection and IP geolocation; acquired by Quova, Inc. in Nov. 2006 and currently being tested at large Internet services. `http://illuminati.coralcdn.org/`
2006–present	**Enterprise networks.** Design and implementation contributions to Ethane [2] [**??**], a backwards-compatible protection and management architecture for enterprise networks. Ethane network switches provide connectivity through on-demand virtual circuits, yet they enforce security policies on a per-flow basis through centrally-managed, atomic, auditable name bindings. Deployment at Stanford since Nov. 2006, serving hundreds of hosts. `http://yuba.stanford.edu/ethane/`
2005–present	**Reliable email.** Designed and implemented the security and privacy protections in Re:, an email acceptance system that leverages social proximity for automated whitelisting [4], using private matching [9]. Recent analysis of privacy for social networks led to more efficient protocols based only on symmetric-key operations (or achieving stronger properties using bilinear maps) [13].
2005–present	**Fault-tolerance groups.** Researched abstractions for the scalable construction of fault-tolerant, distributed systems [14]. Ongoing work with L. Subramanian on partitioning large, dynamic systems into smaller groups, which apply fault-tolerance or reliable communication protocols.
2000–present	**Privacy-preserving protocols.** Developed cryptographic protocols for private matching (PM), which computes the set intersection between two or more parties' inputs [9]. PM uses the properties of homomorphic encryption to privately evaluate a polynomial representation of input sets. Subsequent work led to improved constructions for keyword search (KS) based on oblivious pseudorandom functions [7]. Earlier research included the design and implementation of a prototype system for anonymous cryptographic e-cash (with S. Brands and I. Goldberg), as well as considerations for privacy-enabled digital rights management (DRM) systems [19] [22].
2000–2002	**Anonymity systems.** Designed and implemented Tarzan [12] [20], a peer-to-peer anonymous IP network layer that is strongly resistant to traffic analysis. Helped design Free Haven, a distributed system for the anonymous publishing, storage, and retrieval of information [23] [24] [27].

Positions

3/06–present	**Co-founder** (with Martin Casado). Illuminics Systems, Mountain View, CA.
9/05–present	**Research Assistant.** Stanford University (SCS Group), Stanford, CA.
5/05–8/05	**Research Assistant.** University of California, Berkeley, Berkeley, CA.
9/02–5/05	**Research Assistant.** New York University (SCS Group), New York, NY.
5/03–8/03	**Research Associate.** HP Labs (Trusted Systems Lab), Princeton, NJ.
9/01–6/02	**Research Assistant.** MIT LCS (PDOS Group), Cambridge, MA.
5/01–8/01	**Research Intern.** InterTrust Technologies (STAR Lab), Santa Clara, CA.
6/00–8/00	**Research Intern.** Zero-Knowledge Systems Labs, Montreal, Quebec.
2/99–5/01	**Undergrad Researcher.** MIT LCS (SLS and CIS Groups), Cambridge, MA.
6/99–8/99	**Intern.** Sun Microsystems (HPC Group), Burlington, MA.
6/98–8/98	**Intern.** Cognex Corporation, Natick, MA.
6/96–2/98	**Undergrad Researcher.** MIT Francis Bitter Magnet Lab, Cambridge, MA.

Service

5/03–5/05	**Founder and Organizer.** NYU Systems Reading Group, New York, NY.
2/04–5/05	**Faculty Representative.** NYU Courant Student Organization, New York, NY.
9/01–5/02	**Co-organizer.** MIT Applied Security Reading Group, Cambridge, MA.
9/97–5/02	**President, VP, Winter School Organizer.** MIT Outing Club, Cambridge, MA

Teaching

1/04–5/04	**Teaching Assistant, Lab Instructor.** V22.0480—Computer Networks, NYU.
2/02–5/02	**Teaching Assistant.** 6.033—Computer System Engineering, MIT.
2/01–5/01	**Teaching Assistant.** 6.033—Computer System Engineering, MIT

Advising

Masters	Justin Pettit (Stanford), Robert Soule (NYU), Jeff Borden (NYU)
Undergraduates	Jeffrey Spehar (Stanford), Kevin Shanahan (NYU), Ed Kupershlak (NYU)

Professional activities

Program comm.	WORLDS ’06, UPGRADE-CDN ’06, IRIS Student P2P Workshop ’03
External reviews	NSDI ’07, LATIN ’06, HotNets ’05, EUROCRYPT ’05, Usenix Technical ’05, ISC ’04, CRYPTO ’04, IPDPS ’04, INFOCOM ’04, CCS ’03, SOSP ’03, ISC ’03, PODC ’03, EUROCRYPT ’03, WPES ’02
Journal reviews	ACM Transactions on Computer Systems (TOCS), Journal of Cryptology, Journal of Parallel and Distributed Computing (JPDC), Handbook of Internet Security - P2P Security (Wiley & Sons), Computer Journal

Honors

NDSEG (DoD) Graduate Fellow, 2002-2005
NYU McCracken Fellow, 2002-2006
Henning Biermann Award, NYU Computer Science, 2005 (for outstanding education and service)

Best demo (OASIS), WORLDS 2005.
First paper (highest-ranked), EUROCRYPT 2004 [9].
Award paper, CCS 2002 [12].

Awarded NSF Graduate Fellowship, 2001
Awarded Gordon Wu Fellowship (Princeton), 2001 ; Sterling Prize Fellowship (Yale), 2001
Awarded Graduate Fellowships (U.C.Berkeley, Carnegie-Mellon, UCSD), 2001

Coca-Cola Scholar, 1997-2001 ; Tylenol Scholar, 1997-1999 ; Big 33 Scholar, 1997-1998
Tau Beta Pi, 2000 ; Eta Kappa Nu, 2000 ; Sigma Xi, 2000 ; Order of Omega, 1999
Congressional Award, Silver (1996) and Bronze (1993) medals

Refereed conference publications

[1] Martin Casado and **Michael J. Freedman**. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *Proc. 4th Symposium on Networked Systems Design and Implementation (NSDI 07)*, Cambridge, MA, April 2007.

[2] Martin Casado, Tal Garfinkle, Aditya Akella, **Michael J. Freedman**, Dan Boneh, Nick McKeown, and Scott Shenker. SANE: A protection architecture for enterprise networks. In *Proc. 15th USENIX Security Symposium*, pages 137–151, Vancouver, BC, August 2006.

[3] **Michael J. Freedman**, Karthik Lakshminarayanan, and David Mazières. OASIS: Anycast for any service. In *Proc. 3rd Symposium on Networked Systems Design and Implementation (NSDI 06)*, pages 129–142, San Jose, CA, May 2006.

[4] Scott Garriss, Michael Kaminsky, **Michael J. Freedman**, Brad Karp, David Mazières, and Haifeng Yu. Re: Reliable email. In *Proc. 3rd Symposium on Networked Systems Design and Implementation (NSDI 06)*, pages 297–310, San Jose, CA, May 2006.

[5] **Michael J. Freedman**, Mythili Vutukuru, Nick Feamster, and Hari Balakrishnan. Geographic locality of IP prefixes. In *Proc. 5th ACM SIGCOMM Conference on Internet Measurement (IMC 05)*, pages 153–158, Berkeley, CA, October 2005.

[6] Siddhartha Annapureddy, **Michael J. Freedman**, and David Mazières. Shark: Scaling file servers via cooperative caching. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI 05)*, pages 129–142, Boston, MA, May 2005.

[7] **Michael J. Freedman**, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom function. In *Proc. 2nd Theory of Cryptography Conference (TCC 05)*, pages 303–324, Cambridge, MA, February 2005.

[8] Yevgeniy Dodis, **Michael J. Freedman**, Stanislaw Jarecki, and Shabsi Walfish. Versatile padding schemes for joint signature and encryption. In *Proc. 11th ACM Conference on Computer and Communication Security (CCS 04)*, pages 344–353, Washington, D.C., October 2004.

[9] **Michael J. Freedman**, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology — EUROCRYPT 2004*, pages 1–19, Interlaken, Switzerland, May 2004.

[10] Maxwell Krohn, **Michael J. Freedman**, and David Mazières. On-the-fly verification of rateless erasure codes for efficient content distribution. In *Proc. IEEE Symposium on Security and Privacy*, pages 226–240, Oakland, CA, May 2004.

[11] **Michael J. Freedman**, Eric Freudenthal, and David Mazières. Democratizing content publication with Coral. In *Proc. 1st Symposium on Networked Systems Design and Implementation (NSDI 04)*, pages 239–252, San Francisco, CA, March 2004.

[12] **Michael J. Freedman** and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proc. 9th ACM Conference on Computer and Communications Security (CCS 2002)*, pages 193–206, Washington, D.C., November 2002.

Refereed workshop publications

[13] **Michael J. Freedman** and Antonio Nicolosi. Efficient private techniques for verifying social proximity. In *Proc. 6th International Workshop on Peer-to-Peer Systems (IPTPS 07)*, Bellevue, WA, February 2007.

[14] **Michael J. Freedman**, Ion Stoica, David Mazières, and Scott Shenker. Group therapy for systems: Using link-attestations to manage failures. In *Proc. 5th International Workshop on Peer-to-Peer Systems (IPTPS 06)*, Santa Barbara, CA, February 2006.

[15] **Michael J. Freedman**, Karthik Lakshminarayanan, Sean Rhea, and Ion Stoica. Non-transitive connectivity and DHTs. In *Proc. 2nd Workshop on Real, Large, Distributed Systems (WORLDS 05)*, pages 55–60, San Francisco, CA, December 2005.

[16] Kevin Shanahan and **Michael J. Freedman**. Locality prediction for oblivious clients. In *Proc. 4th International Workshop on Peer-to-Peer Systems (IPTPS 05)*, pages 252–263, Ithaca, NY, February 2005.

[17] Max Krohn and **Michael J. Freedman**. On-the-fly verification of erasure-encoded file transfers (extended abstract). In *Proc. 1st IRIS Student Workshop on Peer-to-Peer Systems*, Cambridge, MA, August 2003.

[18] **Michael J. Freedman** and David Mazières. Sloppy hashing and self-organizing clusters. In *Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS 03)*, pages 45–55, Berkeley, CA, February 2003.

[19] Joan Feigenbaum, **Michael J. Freedman**, Tomas Sander, and Adam Shostack. Economic barriers with existing privacy technologies in e-commerce systems. In *Proc. Workshop on Economics and Information Security*, Berkeley, CA, May 2002.

[20] **Michael J. Freedman**, Emil Sit, Josh Cates, and Robert Morris. Introducing Tarzan, a peer-to-peer anonymizing network layer. In *Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS 02)*, pages 121–129, Cambridge, MA, March 2002.

[21] **Michael J. Freedman** and Radek Vingralek. Efficient peer-to-peer lookup based on a distributed trie. In *Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS 02)*, pages 66–75, Cambridge, MA, March 2002.

[22] Joan Feigenbaum, **Michael J. Freedman**, Tomas Sander, and Adam Shostack. Privacy engineering in digital rights management systems. In *Proc. ACM Workshop in Security and Privacy in Digital Rights Management (DRM 01)*, pages 76–105, Philadelphia, PA, November 2001.

[23] Roger Dingledine, **Michael J. Freedman**, David Hopwood, and David Molnar. A reputation system to increase MIX-net reliability. In *Proc. Information Hiding Workshop (LNCS 2137)*, pages 126–141, Pittsburgh, PA, March 2001.

[24] Roger Dingledine, **Michael J. Freedman**, and David Molnar. The Free Haven Project: Distributed anonymous storage service. In *Proc. Workshop on Design Issues in Anonymity and Unobservability (LNCS 2009)*, pages 67–95, Berkeley, CA, July 2000.

In submission

[25] Martin Casado, **Michael J. Freedman**, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. Ethane: Taking control of the enterprise, 2007.

Unrefereed publications, book chapters

[26] **Michael J. Freedman**. Automating server selection with OASIS. In *;login: The USENIX Magazine*, pages 46–52, October 2006.

[27] Roger Dingledine, **Michael J. Freedman**, David Molnar, and David Parkes. Reputation. In *Digital Government Civic Scenario Workshop*, Cambridge, MA, April 2003.

[28] Roger Dingledine, **Michael J. Freedman**, and David Molnar. *Peer-to-Peer: Harnessing the Power of Disruptive Technology*, chapter Accountability, pages 271–340. O'Reilly, 2001.

[29] Roger Dingledine, **Michael J. Freedman**, and David Molnar. *Peer-to-Peer: Harnessing the Power of Disruptive Technology*, chapter Free Haven, pages 159–190. O'Reilly, 2001.

References

Prof. David Mazières
Stanford University
Computer Science Department
353 Serra Mall, #290
Stanford, CA 94305-9025
(650) 723-8777
`rec@nospam.scs.stanford.edu`

Prof. Frans Kaashoek
Massachusetts Institute of Technology
Stata Center, #32-G992
77 Massachusetts Avenue
Cambridge, MA 02139
(617) 253-7149
`kaashoek@csail.mit.edu`

Prof. Ion Stoica
University of California, Berkeley
RADLab, Room 465
Soda Hall #1776
Berkeley, CA 94720-1776
(510) 643-4007
`istoica@cs.berkeley.edu`

Prof. Nick McKeown
Stanford University
Computer Science Department
353 Serra Mall, #340
Stanford, CA 94305-9025
(650) 725-3641
`nickm@stanford.edu`

Prof. Joan Feigenbaum
Yale University
Computer Science Department
P.O. Box 208285,
New Haven, CT 06520-8285
(203) 432-6432
`joan.feigenbaum@yale.edu`

Stanford, CA, January 31, 2007

Research statement

Michael J. Freedman

My research interests span the areas of distributed systems, security, networking, and cryptography. I particularly enjoy devising technologies that make new functionality broadly available. My work generally tackles systems problems by coupling principled designs with real-world deployments.

A common thread in my research is the extension of systems designed for centralized or trusted entities into decentralized, untrusted, unreliable, or chaotic settings. These scenarios offer significant challenges, yet they are ones ideally suited for academic research: Such problems or architectures do not naturally arise from within industry, even though the techniques often may be applied back into managed environments, *e.g.*, to survive disasters or to operate safely under attack. More than that, open systems encourage further innovation.

I approach these problems through the innovative use of cryptography, algorithms, or abstractions. By leveraging the resulting properties, one can create self-organizing systems out of unreliable nodes, incentivize proper operation, curtail the impact of malicious behavior, or improve manageability to overcome system brittleness.

Such solutions still require solid engineering, always with the end-user in mind. By providing desired functionality, even research systems can attract users, gain traction, and then truly test the system's mettle. Deployed systems provide real data to direct future design decisions, and they can serve as platforms for otherwise intractable experiments. While much research relies solely on simulation and emulation, only at scale can we truly evaluate many systems—learning from their strengths, weaknesses, and emergent properties—and thus discover new research problems and directions.

Cooperative content distribution. My thesis research focuses on making content delivery more widely available by federating large numbers of untrusted or unreliable machines to share localized resources. Content distribution networks (CDNs) are not a new idea, but the architectures of commercial CDNs are tightly bound to centralized control, static deployments, and cost recovery.

My initial system, CoralCDN [1], explores how to build a self-organizing cooperative web CDN using unreliable hosts. Through its scalable distributed index, nodes can record and locate data without overloading any node, regardless of a file's popularity or system dynamics [1, 2]. Decentralized clustering algorithms enable nodes to find nearby data without querying more distant machines.

CoralCDN incorporates a number of engineering mechanisms for sharing resources fairly and preventing abuse—learned through deployment and community feedback—yet the system is inherently open. Simply modify a URL, and the requested content is automatically retrieved and cached by CoralCDN's proxies. As such, it has been widely adopted in often innovative ways: by servers to dynamically offload flash crowds, by browser extensions to recover from server failures, by podcasting and RSS software, and by daily links on Slashdot and other portals. CoralCDN currently handles about 25 million requests daily from over one million clients.

One challenge in designing CoralCDN was how to compel our unmodified clients to use nearby, unloaded proxies. While commercial systems also deploy *anycast* to select servers, their techniques need handle only a single deployment, often comprised of a mere handful of data centers. Ideally, one public infrastructure could provide anycast for many far-flung services, such that the more services that use it, the more accurate its server-selection results and the lower the bandwidth cost per service.

I built a subsequent system, OASIS [3], that does exactly this: OASIS currently provides anycast among thousands of servers from more than a dozen distributed systems, from both the academic and open-source communities. It flexibly supports a variety of interfaces—currently DNS, HTTP, and RPC—with which clients can discover good servers belonging to the requested system. OASIS can do so because it tackles several problems simultaneously: using nodes from participating services to perform network measurement, detecting and disambiguating erroneous results, representing locality stably across time and deployment changes, and scalably managing state information about many services.

This success at building content delivery from unreliable resources raised the question as to whether we could extend this approach to mutually distrustful clients. Shark [4] provides a distributed file system that improves scalability and performance through cooperative reads, using Coral's indexing layer to locate content. Still, Shark preserves traditional semantics and security: End-to-end cryptography ensures that clients need not trust one another.

We also considered security mechanisms for hosts using rateless erasure codes for cooperative large file distribution. Unfortunately, these codes cannot use traditional authenticators (*e.g.*, hash trees) that guarantee the integrity of individual blocks. Therefore, we devised a homomorphic hash function that can be used to verify downloaded blocks on-the-fly, thus preventing malicious participants from polluting the network with garbage [5]. Implementation aspects mattered in this seemingly-theoretical project. The batching of public-key operations was needed to achieve fast verification, while disk-read strategies led to encoding speeds that even exceeded those of hash trees for non-rateless codes. Finally, for preventing pollution in these non-rateless codes, we showed how simple implementation changes could replace others' heavyweight *black-box* mechanisms.

Recently, I have returned to the problem of moving CoralCDN from its current deployment on PlanetLab onto fully untrusted nodes, as CoralCDN's success has led to bandwidth usage that has long saturated PlanetLab's available capacity. As digital signatures can guarantee content integrity, the challenge is ensuring that sufficient capacity exists. Our latest design promotes resource sharing through incentive-compatible mechanisms: Contributing nodes receive better quality-of-service when the system is under-provisioned. The system applies market pricing techniques to efficiently use available bandwidth, but also incorporates network costs to "play friendly" with service providers. Malicious parties cannot cheat as lightweight cryptographic currency accurately tracks nodes' contributions.

While most of my work on cooperative content distribution has focused on leveraging unreliable or untrusted resources, I am not rigid in my approach. Indeed, some of these systems use logically-centralized components, such as the core OASIS infrastructure or, for each file collection in this last system, servers that manage file prices and currency exchange. Rather, I look where it is sensible or economical to leverage available resources—*e.g.*, local bandwidth for CDNs or measurement points for anycast—and architect systems accordingly. Indeed, these same cost arguments are behind industry's increased interest in such architectures, albeit without the same consideration for security.

Securing decentralized systems. When large decentralized systems lack the necessary security mechanisms, things eventually go awry. The Internet's inter-domain routing protocols (BGP) lack source authentication and thus routes have been hijacked, a weakness shared by DNS. Persistent email spam is frustrating, while false positives from spam filters have made email unreliable. Centralized solutions are not the only answer, however.

Tackling the spam false-positive problem, Re: [6] uses proximity in a social network as a basis for auto-whitelisting email. This approach appears promising given our analysis of large email corpora. And by incorporating our cryptographic protocols for private matching [7, 8], Re: ensures that two parties can maintain privacy without third-party intervention.

In a similar vein, websites want to securely identify their users, but ubiquitous client authentication does not exist. Thus, sites often use weaker identifiers such as IP addresses for access-control decisions, even though edge technologies (NATs, proxies, and DHCP) occlude a server's view of its clients. By instrumenting CoralCDN, we used active web content to measure and analyze the characteristics of over 7 million clients; our results help quantify when and how Internet services can use IP addresses and related information to identify clients [9]. (In fact, our techniques for real-time proxy detection and geolocation were acquired by a leading IP analytics company [10].) Here we see how a system, once widely used, can become a vehicle for otherwise infeasible research. Indeed, we are starting to investigate advertisement click fraud using this platform.

Enterprise networks similarly lack comprehensive security "from the ground up." Instead, a bewildering array of mechanisms (firewalls, NATs, and VLANs) have been retrofitted over the years, leading to brittle, inflexible networks. Begun as a clean-slate design [11], Ethane provides a backwards-compatible protection and management architecture for enterprise networks, where switches establish virtual circuits per flow, after using a domain controller to enforce security policies. Because Ethane simplifies so many network management tasks—testing new policies, deploying new appliances or topologies, performing forensics or fault diagnosis, establishing network isolation classes—its architecture empowers innovation and change within networks. I am further interested in extending such techniques to the wider area for managing autonomous systems.

Future work. Given the challenges of securing and managing networked systems, I have begun to think about new ways to simplify this task.

How can we determine when, where, and why performance or persistent faults in distributed systems occur? I intend to explore lightweight distributed tracing to track transactions across hosts and within processes. By tainting network communication and annotating code, we can generate system-wide "call graphs" during run-time. Of particular interest are identifying normal and anomalous system behavior, possibly through machine learning, and building feedback loops for automated reconfiguration. Other approaches to fault monitoring, detection, and diagnosis may be similarly promising. Of course, having deployed systems to test such tools is a critical advantage to experience the vagaries of failures in production environments. (In fact, others have used CoralCDN for exactly this [12].)

What new abstractions can provide better reliability in the face of failures? I am currently thinking about how to partition large systems into smaller groups, which can then apply heavyweight fault-tolerance or detection protocols [13]. (Such partitioning appears necessary for scalability.) While handling malicious parties in dynamic settings presents many difficult problems, the goal remains for better operation on faulty resources.

Finally, what privacy-preserving technologies can promote greater information sharing? Researchers, operators, and end-users can all benefit from greater access to data, whether inter-domain routing policies for traffic engineering, patient records for medical research, census and other polling data for the social sciences, or social information for cooperative filtering [6]. Unfortunately, privacy concerns often limit data availability, leading to suggestions such as private matching [7] for merging terrorist watch lists [14]. Yet current general-purpose cryptographic solutions are too inefficient for large datasets, while statistical methods are often not sound. I am interested in leveraging specific application contexts to build better protocols (as done in [8]), as well as exploring interface and architecture design for privacy-preserving systems.

While technology trends may incrementally improve system performance, new techniques are needed to enhance security, scalability, reliability, and manageability. I tackle these problems by applying methods from cryptography, distributed algorithms, game theory, and other principled sources. But real solutions require real testing: My research will embrace both strong design and engineering components, even as new problems arise over time. This unusual dual approach already has enabled my research systems to provide tens of millions of people with their Internet fix, often in surprising ways. Through such deployments we can discover new problems, encourage further innovation, and ultimately make new functionality broadly available.

References

[1] **M. Freedman**, E. Freudenthal, and D. Mazières. Democratizing content publication with Coral. In *Proc. Networked Systems Design and Implementation (NSDI)*, pages 239–252, Mar 2004.

[2] **M. Freedman** and D. Mazières. Sloppy hashing and self-organizing clusters. In *Proc. International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 45–55, Feb 2003.

[3] **M. Freedman**, K. Lakshminarayanan, and D. Mazières. OASIS: Anycast for any service. In *Proc. NSDI*, pages 129–142, May 2006.

[4] S. Annapureddy, **M. Freedman**, and D. Mazières. Shark: Scaling file servers via cooperative caching. In *Proc. NSDI*, pages 129–142, May 2005.

[5] M. Krohn, **M. Freedman**, and D. Mazières. On-the-fly verification of rateless erasure codes for efficient content distribution. In *Proc. IEEE Security and Privacy*, pages 226–240, May 2004.

[6] S. Garriss, M. Kaminsky, **M. Freedman**, B. Karp, D. Mazières, and H. Yu. Re: Reliable email. In *Proc. NSDI*, pages 297–310, May 2006.

[7] **M. Freedman**, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology — EUROCRYPT 2004*, pages 1–19, May 2004.

[8] **M. Freedman** and A. Nicolosi. Efficient private techniques for verifying social proximity. In *Proc. IPTPS*, Feb 2007.

[9] M. Casado and **M. Freedman**. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *Proc. NSDI*, Apr 2007.

[10] Quova. http://www.quova.com/, 2006.

[11] M. Casado, T. Garfinkle, A. Akella, **M. Freedman**, D. Boneh, N. McKeown, and S. Shenker. SANE: A protection architecture for enterprise networks. In *Proc. USENIX Security Symposium*, pages 137–151, Aug 2006.

[12] P. Reynolds, J. Wiener, J. Mogul, M. Aguilera, and A. Vahdat. WAP5: Black-box performance debugging for wide-area systems. In *Proc. WWW*, May 2006.

[13] **M. Freedman**, I. Stoica, D. Mazières, and S. Shenker. Group therapy for systems: Using link-attestations to manage failures. In *Proc. IPTPS*, Feb 2006.

[14] J. Dempsey and P. Rosenzweig. Technologies that can protect privacy as information is shared to combat terrorism. Heritage Foundation Legal Memo #11, May 26 2004.

Teaching statement

Michael J. Freedman

My greatest joy in teaching is helping passionate, hard-working students gain the appropriate tools, knowledge, and skepticism to become independent thinkers and researchers. Given my research interests, this largely translates to sharing my enthusiasm for tackling challenging systems problems and building complete solutions. Designing and building systems requires a broad background in understanding potential approaches, recognizing design tradeoffs, and recalling past successes and failures, much of which can be learned through coursework. But equally critical is the judgment one gains from *doing*: conceptualizing the interplay of various system components, approaches, and often devilish details, and identifying a system's shortcomings through analysis in order to improve it.

My goal, both as an advisor and as a teacher, is to empower students to make their own design decisions and, ultimately, to discover their own interesting problems to tackle. During graduate school, I had the opportunity to supervise research projects for six students, both masters and undergraduates. The challenge was to offer well-defined problems when students needed more supervision, sometimes proposing one or more promising approaches and incremental milestones. Still, I found it important to maintain some vision or open-ended problems that students could work towards.

The students' research experiences helped lead some of them to pursue further graduate education (Robert Soule is now a PhD student at NYU), while it gave others their first experience at writing academic papers (Kevin Shanahan was the first author of a workshop paper on peer-to-peer localization). The most successful outcomes emerged from situations where students ultimately were excited by their research and identified their concrete contributions. Especially motivating were projects that impacted a large audience, *e.g.*, one student built a data collection infrastructure for CoralCDN, knowing that his code would touch data from tens of millions of users. My personal experience has been very similar: My academic highlights from college were the research projects where I played an important role; my worst time was a summer largely spent hacking makefiles written by physicists over two decades.

Beyond supervising independent research, I similarly enjoy teaching students within the classroom setting. I first served as a teaching assistant for the core "Computer System Engineering" course at MIT for two consecutive years. Unfortunately, TAs traditionally only played the role of holding office hours and grading assignments for this course, as faculty taught even recitation sections. Thus, in my second year, I proposed holding an additional weekly small-group tutorial section to help students better learn course concepts and readings—as well as to allow TAs to actually teach—a practice still being done five years later. At NYU, I served as the teaching assistant, lab instructor, and occasional lecturer for the new advanced undergraduate course "Computer Networks," which coupled system programming assignments with academic readings. I also organized and helped teach the MIT Outing Club's month-long winter mountaineering course, which attracted nearly 100 participants. While not academic in nature, the time-intensive experience was gratifying both from my ability to educate others (here, literally, on how to survive) and from deepening my own knowledge in the process. This class, much like project courses, focused on doing, not only on knowing.

Given my research background, I am qualified to teach a variety of courses, including distributed systems, operating or storage systems, security and cryptography, networking, or even software engineering. I am also excited to hold more advanced graduate courses or seminars related to my research areas. I am a strong proponent of project-heavy classes for both advanced undergraduate and graduate students; these go directly towards "hands-on" systems experience and often provide a useful segue into further research.

Finally, I believe that seminars and reading groups play an important role both in staying abreast with the latest research and in learning how to evaluate it critically. At MIT, I helped co-organize an applied security reading group. At NYU, I began a weekly systems seminar and organized it for two years, inviting both outside speakers to present their research and internal volunteers to present others' work. I also served as a student representative at NYU CS faculty meetings, gaining important insight into the concerns and wants of both students and faculty, as well as helping to recruit both new students and faculty to the growing department. The NYU computer science department recognized my contributions with the Henning Biermann award for "outstanding contributions to education and service to the department."

My research statement mentions that I think even academic systems should be user-centric. I am similarly drawn to the eminently "user-centric" nature of teaching. After all, professors ultimately are tasked with producing both research *and* students.