Theory of Networked Computing

The increasing prominence of the Internet, the Web, and large data networks in general has profoundly affected social and commercial activity. It has also wrought one of the most profound shifts in Computer Science since its inception. Traditionally, Computer-Science research focused primarily on understanding how best to design, build, analyze, and program computers. Research focus has now shifted to the question of how best to design, build, analyze, and operate networks. How can one ensure that a network created and used by many autonomous organizations and individuals functions properly, respects the rights of users, and exploits its vast shared resources fully and fairly?

The SIGACT community can help address the full spectrum of research questions implicit in this grand challenge by developing a Theory of Networked Computation (ToNC), encompassing both positive and negative results. Algorithms and complexity-theory research has already evolved with and influenced the growth of the Web, producing interesting results and techniques in diverse problem domains, including search and information retrieval, network protocols, error correction, Internet-based auctions, and security. A more general Theory of Networked Computation could influence the development of new networked systems, just as formal notions of "efficient solutions" and "hardness" have influenced system development for single machines. To develop a full-fledged Theory of Networked Computation, researchers will build on past achievements both by striking out in new research directions and by continuing along established directions.

The SIGACT community has identified three broad, overlapping categories of ToNC-research goals:

- **Realizing better networks:** Numerous theoretical-research questions will arise in the design, analysis, implementation, deployment, operation, and modification of future networks.
- **Computing on networks:** Formal computational models of future networks will enable us both to design services, algorithms, and protocols with provable properties and to demonstrate (by proving hardness results) that some networked-computational goals are unattainable.
- **Solving problems that are created or exacerbated by networks:** Not all of the ToNC-research agenda will involve new computational models. The importance of several established theoretical-research areas has risen dramatically as the use of networked computers has proliferated, and some established methods and techniques within these areas are not general or scalable enough to handle the problems that future networks will create. Examples of these areas include massive-data-set algorithmics, error-correcting codes, and random-graph models.

CISE's NetSE program (http://www.nsf.gov/pubs/2008/nsf08578/nsf08578.htm) welcomes proposals in all three categories. For more details about the ToNC-research agenda, see http://www.cs.yale.edu/homes/jf/ToNC.html.

Definitions and Models of Networked Computation

A broad range of theoretical research questions is likely to arise in the design, analysis, implementation, deployment, operation, and modification of future networks. Given our limited ability to model, measure, predict, and control today's Internet, we will need a more principled approach if we are to realize the ambitious goals now under discussion. What are the right primitives and abstractions with which to study networks? How should responsibility for essential network functions be assigned to various network components? How should state be allocated among components? What should the relationships be among naming, addressing, and routing; indeed, which objects in the network should have names that are meaningful network-wide?

In the systems-research community, these questions are representative of "network-architecture" research. From the SIGACT-community perspective, this type of question must be answered in the process of formally defining various types of networks and rigorously formulating models of networked computation.

From a ToNC perspective, one of the most basic unanswered questions is exactly what we mean by "a network" and by "networked computation." Clearly, networks have been in use for quite a while, and some of their computational capabilities and limitations have been formalized. However, existing definitions and models are not precise or comprehensive enough to enable us to prove the type of rigorous, general theorems about what can and cannot be computed on various sorts of networks that would constitute a rich and powerful "Theory of Networked Computation." Part of the difficulty is that the notion of a network has been a moving target, with new types of networks (such as sensor nets and wireless networks) gaining in prominence, making formal definitions a challenge. Our experience with networks is now sufficiently advanced that this difficulty can be overcome.

> **Research Goal:** Formulate the definition(s) that a computational system must satisfy if it is to be called a "network." Which critical resources are consumed in networked computation, and what upper bounds on the consumption of these resources must be satisfied for a networked computation to be considered "efficient"? Formulate notions of "reduction" that can be used to prove that one networked-computational problem is at least as hard as another or that two such problems are equivalent. Identify natural network-complexity classes and problems that are complete for those classes.

Multiple definitions and formal models may be needed, because "future networks" means more than just "next-generation Internet." The ToNC scope will also include theoretical aspects of the DoD's Global Information Grid [GIG], sensor networks, MANETS[1], closed "enterprise" networks, *etc*. Should each type of network be formulated independently, or is there one basic model with a few key parameters? What are the key properties that these parameters would have to capture? Open and evolving vs. closed and stable? Mobile vs. stationary? Designed vs. observed? Homogeneous vs. heterogeneous? Controllable vs. emergent? Is there a formal theory in which all of

---

[1] "MANET" stands for Mobile Ad-hoc NETwork.

these network types are actually distinct, and how does one prove that a given computational system falls into one particular category but not another?

These questions may seem overly ambitious, but similar theoretical frameworks have been developed and have proven highly useful in the related areas of parallel and distributed computing; examples include various PRAM models [Harr, Vish], Valiant's BSP model [Vali], the LogP model [CKP+], and Byzantine error models [LPS] .

> **Research Goal:** Develop a taxonomy of networks, with the goals of categorizing the important computational tasks that can and cannot be done efficiently on each network class and of classifying practical network designs.

References

[CKP+] D. Culler, R. Karp, D. Patterson, A. Sahay, K. Schauser, E. Santos, R. Subramonian, T. Eicken, "LogP: towards a realistic model of parallel computation," in *Proceedings of the 4th Symposium on Principles and Practice of Parallel Programming*, ACM Press, New York, 1993, pp. 1-12.

[GIG] Global Information Grid, http://www.globalsecurity.org/intell/systems/gig.htm

[Harr] T. Harris, "A Survey of PRAM Simulation Techniques," *ACM Computing Surveys* **26** (1994), pp. 187-206.

[LSP] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems* **4** (1982), pp. 382-401.

[Vali] L. G. Valiant, "A bridging model for parallel computation," *Communications of the ACM* **33** (1990), pp. 103-111.

[Vish] U. Vishkin, "A Case for the PRAM as a Standard Programmer's Model," in *Proceedings of the Workshop on Parallel Architectures and Their Efficient Use: State of the Art and Perspectives* (First Heinz-Nixdorf Symposium), LNCS Volume 678, Springer, Berlin, 1993, pp. 11-19.

Economic Agency in Networked Computation

Multi-agent systems have been extensively studied in both Economics and Computer Science, but the two communities have approached the topic very differently. The Economics literature traditionally stressed incentives and downplayed the design of algorithms and protocols, and the Computer-Science literature traditionally did the opposite. The emergence of the Internet has radically changed this state of affairs: Ownership, operation, and use by many self-interested, independent parties gives the Internet characteristics of an economy as well as those of a computer.

Economic agency appears on many levels in diverse types of networks. Internet domains (aka "autonomous systems" or ASes) are the subnetworks that directly serve users, *e.g.*, those run by companies for their employees, by universities for their students, or for commercial ISPs for their customers. They are organizationally and economically independent of each other (indeed some are direct competitors), and yet they must coordinate in order to enable interdomain communication. Nonetheless, re-examination of the autonomous-system concept is part of the clean-slate design agenda in network-architecture research:

> **Research Goal:** Are autonomous systems an essential part of Internet architecture? Are there more monolithic alternatives that could deliver significant advantages? If autonomous systems are essential, is the current hierarchical autonomous-system [GR] structure optimal?

On another level, individual users are self-interested, and they access networks through general-purpose computers that can be reconfigured in order to improve local performance; hence, network operators have to incentivize behavior that leads to good network-wide performance. In wireless mesh and ad-hoc networks, bandwidth is typically contributed and controlled by individual participating nodes; network performance could suffer dramatically if nodes fail to forward others' traffic in order to conserve local resources and are not penalized for this failure. To some extent, it is the centrality of economic agency that is now distinguishing the study of "networking" from that of parallel or distributed computing. For example, instead of studying agents who deviate from network protocols arbitrarily, as has commonly been done in distributed-systems research, it makes sense to consider agents who deviate from network protocols rationally in order to maximize their own utility.

The SIGACT community has focused intently on incentive issues in recent years, especially on the design of incentive-compatible algorithms. By building explicit payments to computational agents into the protocol, a system designer can incentivize the revelation of relevant private information and the choice of strategies that drive the overall system into a desirable equilibrium state. Substantial progress has been made in the design of incentive-compatible protocols for routing, multicast cost sharing, Internet-based auctions, peer-to-peer file distribution, and numerous other problems, but many questions remain open. General questions that form an important part of the ToNC agenda include:

> **Research Goal:** Can one agent determine, through observation, modeling, and data analysis, whether another agent is responding to incentives or rather is behaving "irrationally" in the economic sense of this term?

> **Research Goal:** Can incentive-compatible system designs handle agents with rapidly changing and apparently self-contradictory motivations and utility functions?

> **Research Goal:** Are existing equilibrium concepts (such as strategyproofness, Nash, Bayes Nash, and ex-post Nash), together with randomized and approximate variations put forth recently, sufficient for the analysis of Internet-based computation, or are new, more fundamentally computational definitions needed?

> **Research Goal:** Are standard algorithms concepts compatible with incentive analysis of networked computation? For example, because nodes and links fail, recover, join, and leave large networks frequently, the notion of a single problem instance on which a protocol either does or does not converge and, if it does, converges to a solution that either is or is not optimal may not be applicable. How should one evaluate incentive compatibility of a protocol that is carried out by a changing set of agents and that may never terminate?

Much of the recent work by the SIGACT community on incentive compatibility is covered in [NRTV].

References

[GR] L. Gao and J. Rexford, "Stable Internet Routing without Global Coordination," *IEEE/ACM Transactions on Networking* **9** (2001), pp. 681-692.

[NRTV] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, **Algorithmic Game Theory**, Cambridge University Press, 2007.

Networked Computation on Massive Data Sets

Robust technological trends (*e.g.*, the ever-decreasing cost of data storage, the ever-increasing ubiquity of computers and networks in daily life, and the accelerating deployment of sensor networks and surveillance systems) have led to an explosion of potentially interesting data. This situation has led people in many fields to observe that fresh thinking is needed about data privacy. The flip side of this observation is that these trends also strain our *algorithmic* ability to understand and use available data. Massive-data-set (MDS) computation will thus be a central theme of the ToNC agenda.

The SIGACT community has already taken up this challenge on multiple fronts. New computational models have been developed, including data streaming [FK+1, FK+2, Muth], external memory and cache obliviousness [ABW], and sampling, spot checking, and property testing [EKK+, GGR]. Applications have already been found in network measurement and monitoring (*e.g.*, [EV]). The emphasis has been on near-linear, linear, or even sub-linear time and/or space requirements, because the standard notions of polynomial time and space are inadequate when data sets are truly massive. Randomization and approximation are essential in many MDS tasks, and the fact that the SIGACT community has studied both in depth for many years will stand it in good stead.

Despite recent progress in MDS computation, much remains to be done. Indeed, no computational aspect of massive data is completely understood, and no concrete problem of interest has yet been completely satisfactorily solved. The Web-searching problem domain perfectly exemplifies both the great progress that has been made and the tough challenges that lie ahead. Who could have imagined a decade ago that the web would grow to its current size of tens of billions of publicly accessible pages and that, moreover, one would be able to search through this vast collection of pages in a split second? Despite these advances, most users have had the experience (all too often!) of searching for things that they have not found or of being unable even to express a query in the languages provided by today's search engines.

> **Research Goal:** Develop search techniques that work for images, video, audio, databases, and other non-text data on the web. Look for peer-produced structure in the web that can support search for non-text data in the same way that link structure [Klei] supports keyword search.

One research area that may greatly improve search but has only recently received attention by the SIGACT community is human-aided computing. Humans naturally provide feedback in many ways that could aid search; indeed, recent proposals (*e.g.*, [AD]) suggest creating games that, as a by-product, provide labels that could aid in the image-searching problem we've already highlighted.

Providing theoretical foundations for human-aided networked computation is a particularly novel ToNC challenge. Many observers have celebrated the "democratization" of the information environment that has been wrought by blogs, wikis, chatrooms, and, underlying it all, powerful search. More human input to the search process will make the information environment even more democratic, but it will also strain the algorithmic and mathematical foundations of correctness and information quality that have traditionally been present in the technological world. Trust, noise, and scalability all play a part in human-aided networked computation, and these words mean different things when applied to humans from what they mean when applied to computers.

> **Research Goal:** Develop the theoretical foundations of human-aided networked computation; in particular, develop algorithms that allow networked computers to leverage and aggregate the results of a massive number of human actions. Explore the power and limitations of increasing human involvement in network-based search.

Generalizing from the search domain, numerous Web-based tasks have massive-graph computations at their core. Progress on MDS algorithmics will be an essential part of the solutions.

> **Research Goal:** Given a massive, evolving graph presented as a stream of edge-insertions and -deletions, are there one-pass, space-efficient algorithms to compute (or approximate) key graph properties, *e.g.*, conductance, eigenvalues, and bad cuts?

References

[ABW] J. Abello, A. Buchsbaum, and J. Westbrook, "A Functional Approach to External Graph Algorithms," *Algorithmica* **32** (2002), pp. 437-458.

[AD] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, New York, 2004, pp. 319-326.

[EKK+] F. Ergün, S. Kannan, R. Kumar, R. Rubinfeld, and M. Viswanathan, "Spot checkers," *Journal of Computer and System Sciences* **60** (2000), pp. 717-751.

[EV] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM Transactions on Computer Systems* **21** (2003), pp. 270-313.

[FK+1] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan, "An Approximate L^1-Difference Algorithm for Massive Data Streams," *SIAM Journal on Computing* **32** (2002), pp.131-151.

[FK+2] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, "On Graph Problems in a Semi-Streaming Model," *Theoretical Computer Science* **348** (2005), pp. 207-216.

[GGR] O. Goldreich, S. Goldwasser, and D. Ron, "Property Testing and Its Connection to Learning and Approximation," *Journal of the ACM* **45** (1998), pp. 653-750.

[Klei] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM* **46** (1999), pp. 604-632.

[Muth] S. Muthukrishnan, "Data Streams: Algorithms and Applications," http://athos.rutgers.edu/~muthu/stream-1-1.ps

Experimental Rigor in Networking Research

Until recently, most mainstream Computer-Science research has dealt with “man-made” or “designed” objects: Hardware and software systems were designed, built, programmed, and studied, using approaches and methods akin to those in engineering and mathematics. Today’s large-scale networks (and even large, complex pieces of software) are in some ways closer to the “found” objects or natural phenomena studied by scientists: Detailed knowledge of the constituent components and processes of such a system is often insufficient for understanding and prediction of the system’s aggregate behavior, because of the scale and complexity of the aggregate and the crucial role of exogenous forces, most notably the behavior of human users and operators. This presents abundant opportunity for mathematical modeling and analysis of network behavior.

One approach to modeling and analysis that has proved fruitful is to divide it into five stages [Mitz]: **Observe** (gather data about the behavior of the network), **Interpret** (explain the importance of these observations in context), **Model** (propose an underlying model for the observed behavior), **Validate** (find data to validate and, if necessary, specialize or modify the model, and **Control** (based on the model, design ways to control the network behavior).

Observation and interpretation have been proceeding apace for many years, and some consistent themes have emerged. For example, power-law and lognormal distributions[1] are observed almost everywhere that there is networked computation, both in Computer Science (file sizes, download times, Internet topology, the Web graph, *etc.*) and in other fields (income distributions, city sizes, word frequency, bibliometrics, species and genera, *etc.*). Despite their ubiquity in the study of network data, we do not yet fully understand how best to use these classes of distributions. In particular, it can be unclear whether observed data are more accurately modeled as a power-law distribution or a lognormal distribution. The distinction can be extremely important in some modeling contexts (*e.g.*, stock prices and insurance tables); when and why it is important in the modeling of network behavior is not always clear.

> **Research Goal:** Develop techniques for distinguishing empirically between power-law distributions and lognormal distributions. For situations in which they cannot be distinguished empirically, explore the implications of both modeling choices for validation of the model and subsequent control of network behavior.

Distinguishing empirically between power-law-distribution models and lognormal-distribution models is a specific case of the validation challenge. In general, there are many models of network behavior in the literature, but there are few effective techniques for validating that a model is the right one in order to predict and control future behavior. Some of the best work on model validation has actually resulted in model refutation [CCG+, LBCX]. Validation is inherently harder than refutation; in fact, it is not clear

---

[1] A *power-law distribution* is one that satisfies $Pr[X \geq x] \sim cx^{-\alpha}$. The random variable X is *lognormally distributed* if $\ln X$ is normally distributed.

exactly what constitutes convincing validation. Fleshing out this area is a basic ToNC challenge.

> **Research Goal:** Develop techniques for validating models of network behavior, *e.g.*, for proving that a probabilistic model is consistent with observed data or that one model is a "better fit" than another.

Ultimately, the goal of network modeling and analysis is the ability to predict and control network behavior. Accurate models should inform the co-design of networks and algorithms. They should also empower us to change various aspects of network design, use, or operation in ways that improve performance without unforeseen negative side-effects. Many other themes explored in this report, *e.g.*, incentive compatibility, network algorithmics, and networked-computational complexity, might be useful for control.

> **Research Goal:** Explore the feasibility of controlling networks for which models have been validated. In particular, explore the use of incentives (both with and without monetary transfers), limits on users' access to network resources (such as space and bandwidth), and limits on access to information about the network state.

Progress toward these goals will require significant advances in experimental networking research and facilities of a type and scale that are currently unavailable.

There are also purely theoretical problems that beckon in the area of analytical paradigms for networked computation. For example, the network analog of *smoothed* analysis [ST] would clearly be useful. Smoothed analysis, which has shed light on classic problems such as the running time of the simplex algorithm for solving linear programs, captures the fact that there can be uncertainty about in the input to an algorithm. This is quite relevant to network algorithms, where the uncertainty might come from, *e.g.*, unpredictable traffic congestion, unreliable network components, unpredictable user behavior, or intentionally supplied random bits.

> **Research Goal:** Expand the scope of network modeling and analysis. In particular, develop holistic models that capture many network features simultaneously and analytical methods that exploit uncertainty about the environment.

References

[CCG+] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The Origin of Power Laws in Internet Topologies Revisited," in *Proceedings of INFOCOM 2002*, IEEE Computer Society Press, Los Alamitos, 2002, pp. 608–617.

[LBCX] A. Lakhina, J. Byers, M. Crovella, and P. Xie, "Sampling Biases in IP Topology Measurements," in *Proceedings of INFOCOM 2003*, IEEE Computer Society Press, Los Alamitos, 2003, pp. 332–341.

[Mitz] M. Mitzenmacher, "Editorial: The Future of Power Law Research," *Internet Mathematics* **2** (2006), pp. 525-534.

[ST] D. Spielman and S. Teng, "Smoothed Analysis: Why the Simplex Algorithm Usually Takes Polynomial Time," *Journal of the ACM* **51** (2004), pp. 385-463.

Theory of Networked Computation: Participants

This material was generated at two ToNC workshops during Spring semester 2006, one at the Nassau Inn in Princeton, NJ on February 16-17 and the other at the International Computer Science Institute (ICSI) in Berkeley, CA on March 16-17. Both workshops were attended by invited participants and by members of the Computer Science community who sent in successful applications. At both events, plenary talks were presented on important ToNC themes, and then participants formed "breakout groups" for in-depth discussion and problem formulation.

The Princeton ToNC workshop was chaired by Joan Feigenbaum and Jennifer Rexford. Breakout-group themes were Next-Generation Information Systems (Andrei Broder, chair), Next-Generation Network Architecture (Ashish Goel, chair), Next-Generation Network Protocols (Bruce Maggs, chair), Control of Personal Information in a Networked World (Rebecca Wright, chair), and Economic Approaches and Strategic Behavior in Networks (Michael Kearns, chair). The participants were Matthew Andrews (Bell Labs), Sanjeev Arora (Princeton), James Aspnes (Yale), Hari Balakrishnan (MIT), Boaz Barak (Princeton), Amotz Barnoy (Brooklyn College, CUNY), Andrei Broder (Yahoo! Research), Moses Charikar (Princeton), Nick Feamster (Georgia Institute of Technology), Joan Feigenbaum (Yale) , Michael Foster (NSF), Ashish Goel (Stanford) , David Goodman (NSF), David Johnson (AT&T Labs) , Howard Karloff (AT&T Labs) , Richard Karp (UC Berkeley and ICSI), Jonathan Katz (University of Maryland), Michael Kearns (University of Pennsylvania) , Vincenzo Liberatore (Case Western Reserve University), Bruce Maggs (CMU and Akamai), Stephen Mahaney (NSF), S. Muthukrishnan (Rutgers), Kathleen O'Hara (NSF), Jennifer Rexford (Princeton), Rahul Sami (University of Michigan), Alex Snoeren (UC San Diego), Daniel Spielman (Yale), William Steiger (NSF), Eva Tardos (Cornell), Robert Tarjan (Princeton), Sirin Tekinay (NSF) , Eli Upfal, (Brown), Avi Wigderson (IAS), Gordon Wilfong (Bell Labs), Tilman Wolf (University of Massachusetts), and Rebecca Wright(Stevens Institute of Technology).

The Berkeley ToNC workshop was chaired by Joan Feigenbaum and Scott Shenker. Breakout-group themes were Algorithmic Foundations of Networked Computing (John Byers, chair), Analytical Foundations of Networked Computing (Eva Tardos, chair), Complexity-Theoretic Foundations of Networked Computing (Russell Impagliazzo, chair), Economic Foundations of Networked Computing (Milena Mihail, chair), and Foundations of Secure Networked Computating (Salil Vadhan, chair). The participants were Moshe Babaioff (SIMS), Kirstie Bellman (Aerospace Corporation), John Byers (Boston University), Chen-Nee Chuah (UC Davis), John Chuang (SIMS), Luiz DaSilva (Virginia Poly), Neha Dave (UC Berkeley), Joan Feigenbaum (Yale), Michael Foster (NSF), Eric Friedman (UC Berkeley [on leave from Cornell]), Joseph Hellerstein (UC Berkeley), Russell Impagliazzo (UC San Diego), Matti Kaariainen (ICSI), Anna Karlin (University of Washington), Richard Karp (UC Berkeley and ICSI), Robert Kleinberg (UC Berkeley/Cornell), Richard Ladner (University of Washington), Karl Levitt (NSF), Gregory Malewicz (Google), Milena Mihail (Georgia Institute of Technology), Christos Papadimitriou (UC Berkeley), Kathleen O'Hara (NSF), Satish Rao (UC Berkeley), Vijay

Raghavan (UC Berkeley), Tim Roughgarden (Stanford), Amin Saberi (Stanford), Scott Shenker (UC Berkeley and ICSI), William Steiger (NSF), Ion Stoica (UC Berkeley), Eva Tardos (Cornell), Shanghua Teng (Boston University), Salil Vadhan (Harvard), and George Varghese (UC San Diego).

Both ToNC workshops were funded by National Science Foundation grant CCF-0601893. Slides for all talks, including breakout-group reports, can be found by following the links on http://www.cs.yale.edu/homes/jf/ToNC.html.