Towards a Theory of Networked Computation

Joan Feigenbaum
http://www.cs.yale.edu/homes/jf

Michael Mitzenmacher
http://www.eecs.harvard.edu/~michaelm

DRAFT

DRAFT – DECEMBER 2006 – DRAFT

This is a draft report that is under revision. Please send comments to Joan.Feigenbaum@yale.edu and MichaelM@eecs.harvard.edu.

Supported by NSF grant CCF-0601893

Executive Summary

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 Theory of Computation (ToC) 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. ToC 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, the ToC community will build on its past achievements both by striking out in new research directions and by continuing along established directions.

Two NSF-sponsored workshops were held during the Spring of 2006 in order to flesh out the ToNC-research agenda [ToNC]. This report contains the results of those workshops. In it, we describe the state of the art of networked computation, some general research themes that constitute the heart of the ToNC scope, specific open problems in ToNC (not an exhaustive list of such problems, but enough to support our claim that progress can be made in this important area by a large segment of the ToC-research community), important issues that cut across multiple research themes, and recommendations for institutional support of ToNC research. Highlights of the report are given here in the Executive Summary, and details can be found in the following sections.

Research Goals

Workshop participants 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.

DRAFT

- **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.

We briefly give the flavor of each category here. Sections II, III, and IV below flesh out in detail the broad-ranging research agenda developed at the workshops [ToNC].

Like today's Internet, future networks may be characterized by massive scale, subnetwork autonomy, user self-interest, device heterogeneity, and/or emergent behavior. 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 better networks." What are the right primitives and abstractions with which to study networks? Is "layering" fundamental, and, if so, what is the optimal set of layers? 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 a ToC perspective, these are the type of questions that must be answered in the process of formally defining various types of networks and rigorously formulating models of networked computation.

With one or more precise definitions of "network" in hand, it will be natural to ask what can be "computed on a network" and how efficiently computations can be done on a network. The Web-searching problem domain perfectly exemplifies both the evidence that networked computation can be tremendously powerful and the tough challenges that lie ahead if it is to be improved. Search engines that handle billions of Web pages and support a dizzying array of economic, scholarly, and social activities are remarkable technological achievements. On the other hand, numerous technical problems (including many of an algorithmic or combinatorial nature) will have to be solved if we are to have "personalized search" (which strongly implicates privacy), defenses against "Google bombing" and other adversarial or strategic behavior by webpage owners, the ability to search for video or audio clips as well as keywords, and many other search capabilities that users clearly want. The existing bodies of theory on parallel and distributed computing may provide partial answers to the questions of what can be "computed on a network" and how efficiently, but the massive scale, subnetwork autonomy, user self-interest, device heterogeneity, and emergent behavior that characterize present and future networks are not satisfactorily dealt with by either of these existing theories.

More generally, a formal complexity-theoretic approach will enable investigation of the *inherent power and limitations* of networked computing. Notions of "resources" and "reductions" will allow us to determine which fundamental networking problems are easy, which are hard, and why. One approach to the development of "complexity theory of networked computation" is the *black-box channel* approach (described in detail in Section III below). In this model, communication channels are described by properties ("bit-hiding channels," "anonymous channels," "authenticated channels," *etc.*); the

DRAFT

composition of two channels is a channel, and thus "protocols" that are themselves channels can be built by composing channels. It may be possible to leverage known reductions among properties to prove both upper and lower bounds on the complexity of protocol-design tasks and to develop a useful notion of "universality" in networked computation (analogous to the notions of universality in circuit computation or Turing-Machine computation).

In the third category ("problems created or exacerbated by networks"), the focus is on scaling up and improving existing models and methods (*e.g.*, streaming, sampling, and sketching) to meet the challenges posed by modern networks. For example, 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? If a (single) computer (that is not a node in the evolving graph under consideration) can compute or approximate these values, can it also efficiently prescribe corrective action when problems are detected?

Cross-Cutting Issues

Several cross-cutting, high-level issues are relevant to all three categories and arose repeatedly during plenary and breakout sessions at both workshops

- **Incentive compatibility:** Perhaps the most important distinguishing feature of modern networks is that they are simultaneously built, operated, and used by multiple parties with diverse sets of interests and with constantly changing mixes of cooperation and competition. Formal models of networked computation and notions of hardness and easiness of computation will have to incorporate subnetwork autonomy and user self-interest in an essential way.
- **SPUR:** Achieving the broadest possible vision of "networked computation" will require substantial progress on Patterson's SPUR agenda [Patt]. In his words, "we have taken ideas from the 1970s and 1980s to their logical extreme, providing remarkably fast and cheap computing and communication (C&C) to hundreds of millions of people. … [F]or our new century, we need a new manifesto for C&C: ... Security, Privacy, Usability, and Reliability (SPUR)."
- **Build on success:** Although today's Internet may leave something to be desired with respect to security, privacy, usability, and reliability, it has far surpassed expectations with respect to several important design goals, *e.g.*, flexibility and scalability. Are the new design criteria compatible with the (manifestly successful) old criteria, and, if not, what are our priorities?
- **"Clean slate":** The phrase "clean-slate design" has become a mantra in networking-research forums and in calls for proposals. Not surprisingly, many people have raised the question of whether anything that requires a "clean slate" could ever be brought to fruition in a world in which networked computation is pervasive and mission-critical. From a research perspective, the crucial point is that clean-slate *design* does not presume clean-slate *deployment*. Part of the ToNC agenda is the evaluation of new technologies, methods, algorithms, *etc.* from the perspective of incremental deployability and paths to adoption.

DRAFT

- **Diversity of "networks":** The scope of the networking research agenda is broader than "next-generation Internet," and thus the ToNC agenda must be broader as well. Interesting theoretical questions arise in the study of special-purpose networks (such as the DoD's Global Information Grid); of moderate-sized but functionally innovative networks; of sensor nets and other technologically constrained networks; of mobile networks; and of P2P and other application-layer networks.

Institutional Support of ToNC

The ToC community will pursue the ToNC-research agenda on many fronts and in many ways. Valuable types of research projects include but are not limited to:

- **Small, single-investigator, purely theoretical projects:** By "small," we mean funded at a level sufficient to pay for one or two months' of PI summer salary per year, one or two PhD students per year, and a few incidentals such as conference travel or commodity computers for the project participants.
- **Medium- and large-sized, multi-investigator projects involving both theory and experimentation:** The distinguishing features of such a project are (1) multiple PIs, at least one of whom is a theorist and at least one of whom is an experimentalist and (2) the inclusion of experimental work on a "real problem" arising in a network that can be built or at least envisioned in the current technological environment. Funding levels for these projects can range from anything that is bigger than "small" to the size of a large-ITR budget.

Program Directors in NSF's Computer and Network Systems Division have explicitly welcomed this type of medium- and large-sized project proposal, and the "distinguishing feature" text above comes from them. Careful consideration was given at the workshops to whether small, purely theoretical projects are equally important for success of the ToNC agenda, and participants decided that they are, for two basic reasons: (1) The intellectual scope of ToNC should not be limited by networks that can be built or even envisioned in the current technological environment; technologically untethered but mathematically rigorous investigation of networked computation is also worthwhile. (2) Some of the most eminent and productive members of the ToC community have traditionally worked by themselves or in collaboration with other theorists, and they have established broad and deep research track records in the process. Some have no experience working closely with experimentalists; nonetheless, they have built theories (*e.g.*, in distributed computing and in cryptography) that are of interest to practitioners as well as theorists. This subcommunity is unlikely to participate if all funded ToNC projects are medium- or large-sized projects of the type described above; yet, its potential contribution to the ToNC agenda is immense and should not be precluded by lack of funding.

DRAFT

Next Steps

For the ToNC-research agenda to be as broad and deep as it promises to be, support will have to be obtained from diverse sources. In particular, funding will have to come from all three divisions in the CISE Directorate at NSF, from Federal agencies other than NSF, and from forward-looking IT companies. This is one of the major challenges ahead for ToNC-community leaders.

Advocacy and outreach will be important in meeting this challenge. The ToNC community will participate in the CS-research community's effort to increase support for computing research in general and for networking research in particular. The role that theory can play in improving our understanding of networked computation will be articulated in white papers, briefing slides, and popular articles. ToNC researchers will continue to promote our technical agenda both in our traditional forums (*e.g.*, STOC, FOCS, SODA, and Complexity) and in forums that unite us with other communities (*e.g.*, EC, PODC, CCS, and NetEcon).

Finally, the ToNC community will coordinate and collaborate with the broader networking community, in advocacy and in research. For example, ToNC researchers can play a vital role in the Global Environment for Network Innovations [GENI] by formulating testable hypotheses about the inherent power and limitations of networks. The architecture-research community is currently wrestling with fundamental questions about the value, costs, and tradeoffs of various networking primitives and abstractions. Very similar questions must be answered in the pursuit of a rigorous Theory of Networked Computation, and GENI would present a unique opportunity to experiment with new networks that have both innovative functionality and rigorous foundations.

DRAFT

I. Introduction

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?

Members of the Theory of Computation (ToC) community held two NSF-sponsored workshops during the Spring of 2006 in order to explore ToC's (ongoing) contribution to research in next-generation networking. Workshop participants presented and developed three broad categories of research challenges in the emerging area of Theory of Networked Computation (ToNC):

- **Realizing better networks:** Numerous theoretical-research questions will arise in the design, analysis, implementation, deployment, operation, and modification of future networks. They are discussed in Section II below.
- **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. They are discussed in Section III below.
- **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. They are discussed in Section IV below.

In fleshing out these three types of ToNC-research challenges, this report has aimed for an intermediate level of specificity. Some explanation is provided for each of the 62 open questions presented, but further problem formulation would be needed on each question before one could hand it off to a beginning graduate student. The hope is that each question or small set of questions will inspire a diverse set of investigators to generate research proposals, either individually or in small teams.

These are three overlapping categories, and thus decisions about which research problems belong in each section are necessarily subjective and imperfect. Indeed, there are several important cross-cutting issues that arise in all three sections:

DRAFT

- **Incentive compatibility:** Perhaps the most important distinguishing feature of modern networks is that they are simultaneously built, operated, and used by multiple parties with diverse sets of interests and with constantly changing mixes of cooperation and competition. Formal models of networked computation and notions of hardness and easiness of computation will have to incorporate subnetwork autonomy and user self-interest in an essential way.
- **SPUR:** Achieving the broadest possible vision of "networked computation" will require substantial progress on Patterson's SPUR agenda [Patt]. In his words, "we have taken ideas from the 1970s and 1980s to their logical extreme, providing remarkably fast and cheap computing and communication (C&C) to hundreds of millions of people. … [F]or our new century, we need a new manifesto for C&C: ... Security, Privacy, Usability, and Reliability (SPUR)."
- **Build on success:** Although today's Internet may leave something to be desired with respect to security, privacy, usability, and reliability, it has far surpassed expectations with respect to several important design goals, *e.g.*, flexibility and scalability. Are the new design criteria compatible with the (manifestly successful) old criteria, and, if not, what are our priorities?
- **"Clean slate":** The phrase "clean-slate design" has become a mantra in networking-research forums and in calls for proposals. Not surprisingly, many people have raised the question of whether anything that requires a "clean slate" could ever be brought to fruition in a world in which networked computation is pervasive and mission-critical. From a research perspective, the crucial point is that clean-slate *design* does not presume clean-slate *deployment*. Part of the ToNC agenda is the evaluation of new technologies, methods, algorithms, *etc.* from the perspective of incremental deployability and paths to adoption.
- **Diversity of "networks":** The scope of the networking research agenda is broader than "next-generation Internet," and thus the ToNC agenda must be broader as well. Interesting theoretical questions arise in the study of special-purpose networks (such as the DoD's Global Information Grid); of moderate-sized but functionally innovative networks; of sensor nets and other technologically constrained networks; of mobile networks; and of P2P and other application-layer networks.

The ToNC-research agenda presented herein is the synthesis of presentations (both invited and contributed) and breakout-group discussions by workshop participants. Detailed information about workshop programs, structure, and attendees can be found in the Appendix below and in [ToNC].

II. Realizing Better Networks

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

DRAFT

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 a ToC 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. Many of the talks and breakout-group discussions at the ToNC workshops were focused on these issues. We summarize some of these discussions in this section.

II.A Formal Models

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.

> **Question II.A.1:** 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.

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 these network types are actually distinct, and how does one prove that a given

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

DRAFT

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] .

> **Question II.A.2:** 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.

Clearly, one of the defining properties of networks is that they provide computational and communication capabilities that permit multiple, physically separate machines to compute something jointly. This property has been the subject of intensive study in the fields of parallel computation and distributed computation, and these existing bodies of theory should have much to contribute to ToNC.

> **Question II.A.3:** Is a "parallel computer" a special case of a "network"? More generally, what are the formal relationships between parallel computation and networked computation and between distributed computation and networked computation?

Part of what distinguishes the study of networked computation from the (more established) studies of parallel and distributed computing is the role of economically and organizationally independent subnetworks. In the Internet, these are the Autonomous Systems (ASes), but the phenomenon is more general. We discuss this aspect of networked computation in more detail in Section II.C below.

An important thing to consider in formulating models of networked computation is simulation and universality. Is there a notion of "universal network" that is analogous to "universal Turing Machine" or "universal circuit" and can play an analogous role in the theory? This question was raised by workshop participants in connection with the Global Environment for Network Innovations [GENI].

> **Question II.A.4:** Will GENI be a "universal network"? That is, will it be able to simulate any system that is properly understood to be a "network"? More concretely, universality would require the ability to realize all legitimate naming, addressing, and routing frameworks; what classes of naming, addressing, and routing frameworks will be realizable in GENI?

We return to the subject of naming, addressing, and routing in Section III below.

II.B Architectural Principles

Basic Internet-design principles are undergoing a critical re-examination in the networking community, supported by various programs in the NSF's Computer and Network Systems Division that are complementary to the GENI initiative. These questions are also central to the formulation of models of networked computation; we briefly examine two of them here.

DRAFT

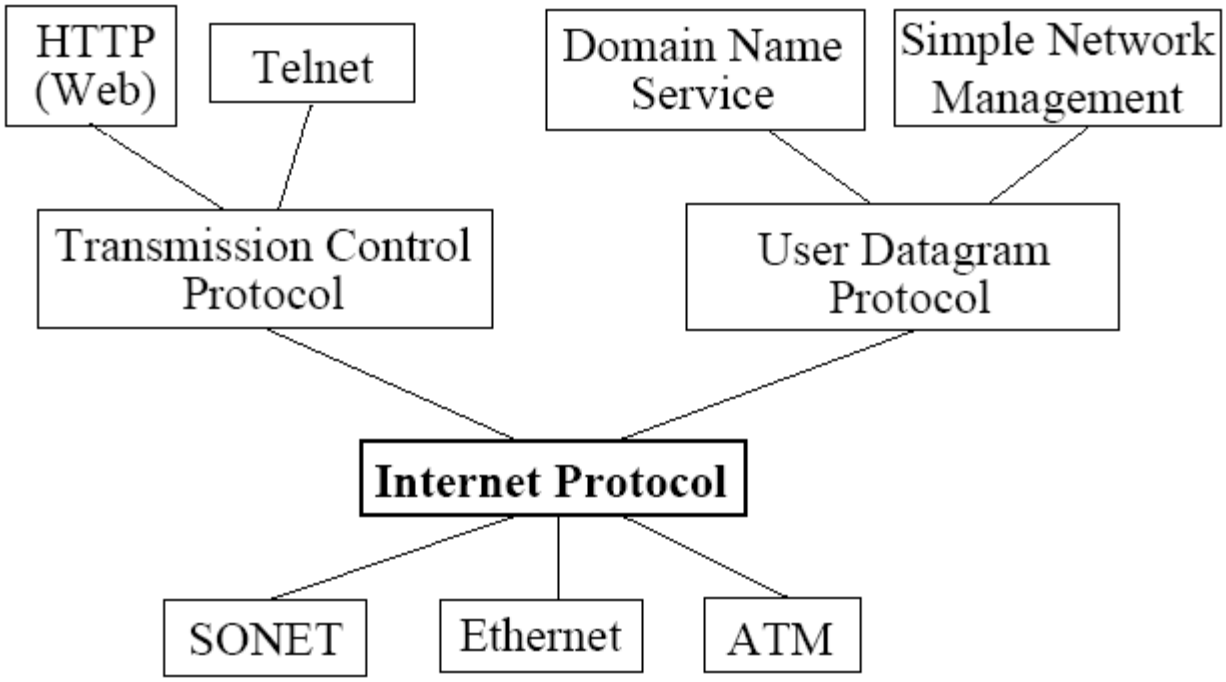
The end-to-end principle is a central tenet of Internet design [SRC]. The network itself is designed merely to transport data from one machine to another. Because basic network services are quite simple and are not "content-aware," they can support a wide range of applications without modification. Moreover, these basic services conform to open protocol standards that anyone can build upon. Users are assumed to access the network through "intelligent endpoints," most notably general-purpose, programmable computers. Inventors and entrepreneurs are free to develop new applications and make them available to Internet users. As long as these applications can communicate via standard Internet protocols, individual "endpoint" owners can just install them and use them; the network itself need not be changed, and thus inventors and entrepreneurs need not go through the typically long, difficult process of designing, implementing, and deploying a modification *of the network infrastructure* in order to deploy a new product that's directed at end users. This basic design principle has led to a platform that is tremendously fertile and dynamic, both technologically and commercially. The most famous demonstration of the power that the end-to-end principle confers is, of course, Tim Berners-Lee's unilateral roll-out of HTTP, HTML, and Mosaic.

Although the end-to-end principle has enabled great things, it has not enabled everything that Internet stakeholders want, and there is mounting evidence that something new might be needed. In a highly influential paper written in the late 1990s (after the Internet, which had been a niche-market platform for more than 20 years, had finally become a mass-market platform), Blumenthal and Clark "conclude that there is a risk that the range of new requirements now emerging could have the consequence of compromising the Internet's original design principles… [They] link this possible outcome to a number of trends: the rise of new stakeholders in the Internet, in particular Internet service providers; new government interests; the changing motivations of a growing user base; and the tension between the demand for trustworthy overall operation and the inability to trust the behavior of individual users" [BC]. Among the most difficult problems to solve without compromising the end-to-end principle are copyright infringement, privacy violation, and inadequate quality-of-service guarantees; these are also among the issues that ToNC workshop participants agreed are important if the next generation's Internet is to be better than the current generation's.

> **Question II.B.1:** Should the end-to-end principle be preserved in its current form, modified, or scrapped altogether in the "clean-slate Internet design" project now underway? What primitives besides transfer can offer the most additional power, while minimizing the impact on the end-to-end principle? How much application-specific functionality in the core of the network is necessary or desirable to meet emerging demands and requirements?

Like the end-to-end principle, the network-architectural principle of *layering* is central to today's Internet and is viewed by many as a crucial enabler of Internet success. The following standard diagram depicts (from bottom to top) the *physical, network, transport*, and *application* layers.

DRAFT



Diverse technologies and protocols have flourished at all layers except for the network layer, in which the Internet Protocol (IP) provides basic naming and addressing capability. If a novel service or device is built to fit into a particular layer L, does not require a change to the layer directly below L, and allows L to continue to provide the necessary functionality to the layer directly above L, it can be deployed without a lengthy, painful redesign of the entire network.

More recent developments in networking have led many people to conclude that the classic, simple layering model is inadequate. Wireless antennas provide one well-known example. Power control in wireless antennas depends on the end-to-end requirements of the application, because the error rate depends on the power, and different applications may have different quality requirements. Furthermore, multiple-antenna radios can (through simulation) present different numbers of channels at different times. The optimal number depends on (and influences) application-level requirements (and properties). Hence, the physical layer has to be aware of the application layer – something that would be precluded by strict enforcement of the layering principle.

Proposed solutions to this type of problem include *cross-layering* and *re-layering*. Cross-layering advocates claim that one should simply allow layers to communicate in certain circumstances, thus violating the orthodox layering philosophy. Re-layering advocates claim that the layering philosophy is sound but that the particular set of layers that has been in use for many years is now outdated; with a new set of layers that is more appropriate for today's networks, cross-layer communication should be avoidable.

DRAFT

Question II.B.2: Formulate a precise definition of "layering," a precise statement of the network-architectural problem that it solves, and criteria by which solutions can be evaluated. Are there alternatives to layering, and how do they compare? In layered network architectures, how can one arrive at an optimal set of layers?

II.C Incentive compatibility

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:

Question II.C.1: 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 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 ToC 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

DRAFT

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 arose at the ToNC workshops include:

Question II.C.2: 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?

Question II.C.3: Can incentive-compatible system designs handle agents with rapidly changing and apparently self-contradictory motivations and utility functions?

Question II.C.4: 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?

Question II.C.5: Are existing ToC 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?

Question II.C.6: Is there a significant role for algorithmic mechanism design without money? For example, when traditional monetary payments cannot be used as short-term or fine-grained incentives (*e.g.*, in battlefield scenarios), can "payments in kind" serve as incentives? The file-sharing system BitTorrent exemplifies this approach: Agents pay for the download bandwidth they consume not with money but rather by providing upload bandwidth to other agents. Can this approach can be generalized, and which system resources can be used as currency in this manner?

Question II.C.7: What types of coalitions should network designers be concerned about, and what types of coalitions are an acceptable and natural part of network development?

Various algorithmic, complexity-theoretic, and mathematical-modeling aspects of incentive compatibility in networks are covered in Sections III and IV below.

DRAFT

II.D Information Privacy, Network Security, and Accountability

Sensitive data abound in today's networked world. By "sensitive data," we mean electronic data records that, if used improperly, can harm data subjects, data owners, data users, or other stakeholders. Personally identifying information (such as names, phone numbers, addresses, social security numbers, and credit-card numbers), copyrighted works, medical records, banking records, and consumer-transaction records are all examples of sensitive data items. It is important to note that some types of sensitive data (*e.g.*, copyrighted works) are not "private" in any colloquial sense of the word and that all may prove to be quite useful for purposes that were not foreseen at the time they were first recorded. The profusion of sensitive data in a variety of public and private network environments and their increasingly pervasive role in everyday life are extremely important developments with wide-ranging social and legal consequences. Very robust technological trends (*e.g.*, the plummeting cost of mass storage and the build-out of broadband networks) ensure that potential misuse will continue to be a central concern for people and organizations.

Taking a system-centric view (instead of the data-centric or user-centric view expressed above) leads to a similarly negative assessment of security and accountability in today's networked world. Inadequate perimeter defenses, denial-of-service attacks, viruses, worms, and other types of malware constantly degrade enterprise-network performance and soak up employee time, and they occasionally disable networks and the organizations supported by them completely. Many attackers are never caught and held accountable for the damage that they cause.

Core design principles of today's Internet, such as the end-to-end principle and the layered architecture discussed in Subsection II.B above, are aimed at moving data packets from one machine to another as simply and quickly as possible. Content-obliviousness of basic communication protocols is a huge asset in the fulfillment of this over-arching goal. Unfortunately, it is also a major obstacle to stakeholder control of data and networks, *e.g.*, to the achievement of user privacy, enterprise-network security, and after-the-fact accountability. Whether one can, by modifying the basic network architecture, preserve the flexibility and scalability of today's Internet while enabling information stakeholders to achieve privacy, security, and accountability is a major open question.

> **Question II.D.1:** Explore network-architectural principles that enable controlled dissemination of personal information and robust protection of network resources. Are these principles consistent with the dynamism, heterogeneity, and scalability present in today's Internet? Formulate precise definitions of these seemingly contradictory goals, and explore the existence of provable, quantifiable trade-offs among them.

Cryptographic theory offers an approach to Question II.D.1: Build into the basic network architecture some of the cryptographic primitives and "set-up assumptions" that are known to be necessary and sufficient for secure multiparty computation without an honest majority, *e.g.*, shared randomness, source authentication, or secure logging [CLOS]. We return to this subject in more detail in Sections III and V below.

DRAFT

If the preventive approaches that have traditionally dominated privacy and security research prove to be inadequate, it might be more productive to rely on accountability mechanisms. Broadly speaking, these are mechanisms that permit people and organizations to keep track of the uses they and others make of sensitive data and machines (together with proof of the efforts they made to obtain authorization) so that they can participate in after-the-fact adjudication procedures.

Today's Internet architecture does not provide adequate support for accountability. Fundamentally, it is based on network addresses, not names. The binding of names to addresses is neither secure nor verifiable, and the same is true of the binding of transmitted data objects to addresses. Consequently, high-level authentication, authorization, and accountability mechanisms that are based on names (as they inevitably will be, because application semantics are expressed in terms of names, not network addresses) can be completely subverted by network-level attacks: denial of service, IP spoofing, DNS spoofing, *etc*.

> **Question II.D.2:** How can one create network resources with universally understood, secure, persistent, verifiable names? What are the minimal sets of securely named network resources upon which one can build proof systems for authorized or accountable access to data and machines? Are there other network-architectural approaches to enabling accountability?

II.E Measurement and Monitoring

Measurement and monitoring are important themes in networking research; to understand today's network and predict the upcoming effects of network usage, one must measure and monitor current happenings accurately and extensively, while minimizing the impact on network performance. *Measurement* tools are used to "inventory" network state, *i.e.*, to determine traffic loads on links, end-to-end packet loss rates, throughput, *etc.* *Monitoring* is the use of measurement to detect events of operational interest, *e.g.*, link failures, "hijacked" routes, apparent intrusions, *etc*. Both give rise to concrete problems of a complexity-theoretic, algorithmic, or combinatorial nature, some of which are addressed in Sections III and IV below. In this section, we pose several general, network-architectural questions in the measurement and monitoring area that are interesting from a ToNC perspective.

Only certain measurements can be made in today's Internet, and they may not be the most useful measurements from a monitoring point of view. For example, threshold-based measurement schemes are fairly easy to implement and are widely used: Count BGP updates, failed TCP connections, or changes in traffic volume. A standard policy is to regard as possibly "anomalous" any counts that exceed certain thresholds. Unfortunately, many events of interest are not obviously characterized by high values of measurable parameters. A single BGP update may signal a route hijack, or a two-byte common substring may encode a polymorphic worm; there may be efficient ways to monitor for such events given the measurement opportunities offered by today's network architecture, but they have not yet been discovered, and threshold schemes in particular will not do the job. In general, the set of operationally interesting events that one may

DRAFT

want to monitor for continues to grow, but the set of network-state measurements that one can make is fixed and does not necessarily suffice for efficient (or even inefficient) detection. Clean-slate Internet design may offer an opportunity to correct this situation, and the ToNC community may be able to contribute.

> **Question II.E.1:** Given a basic network architecture, how should one classify monitoring tasks? Explore the mapping from classes of monitoring tasks to sets of measurement capabilities that enable these tasks to be done efficiently. Explore the inverse of this mapping: Given a set of measurement capabilities, classify the monitoring tasks that these capabilities support.

> **Question II.E.2:** Explore the co-design of functionality and monitorability. Which network architectures are both powerful (in that they provide the functionality and performance that users want) and easy to operate (in that the measurements that operators can make allow them to detect all events of interest)?

As a shared experimental platform, GENI should provide abundant opportunity to test new algorithms, protocols, and modes of analysis. Experimental algorithmic research of this sort presumes the existence of an extensive measurement and monitoring infrastructure, along with well managed data repositories, all of which the ToNC-research community is authorized to use. This leads to two basic questions that were voiced repeatedly at the ToNC workshops:

> **Question II.E.3:** Which data would be most useful for experimental algorithmic research? How can we ensure that these data are collected and that ToNC researchers are authorized to use them?

> **Question II.E.4:** Will steps be taken to ensure adequate security and privacy, so that users of this experimental platform will agree to be monitored?

The crypto-theory community has developed a broad range of techniques for privacy-preserving computation that may be useful in addressing Problem II.E.4. We return to this subject in Section III below.

III. Computing on Networks

Already, almost all computers are connected to at least one network essentially whenever they are on, and an increasing amount of computation now involves two or more machines on a network. Ordinary computation can therefore be reconceptualized to include networked communication between machines as a basic primitive. Which algorithmic and complexity-theoretic definitions best capture this new computational reality? How can we design and deploy next-generation networked computation so that it develops in a safe, secure, and efficient manner? Considerable attention was paid to this class of questions at the ToNC workshops, and we report on those talks and discussions in this section.

DRAFT

III.A Complexity Theory of Networked Computation

ToC research has already evolved with and influenced the growth of the Internet and the World Wide Web. Despite the existence of many interesting results, both positive and negative, in a variety of technical areas, there is as yet no coherent Theory of Networked Computation that allows us to draw stark, provable boundaries between those networked-computational problems that can be solved efficiently and those that cannot. Formal Theory of (single-machine) Computation has given us efficient algorithms for some basic problems, hardness results for other basic problems, and, more generally, a wealth of techniques and intellectual frameworks for formalizing, analyzing, classifying, and solving computational problems. It has also made a lasting contribution to the development of hardware and software systems. The ToC community can build on all of this experience to make a similar or greater contribution to the development of future networks.

For a comprehensive theory, the first step is a computational model in which essential features of the Internet (*e.g.*, massive scale, subnetwork autonomy, user self-interest, device heterogeneity, and emergent behavior) are modeled; this is the thrust of Question II.A.1 above. With one or more such models in hand, we could take the next steps and develop network algorithmics and network-complexity theory. Notions of "resources" and "reductions" would enable us to define complexity classes and to place fundamental networking problems in these classes. Identifying problems that are provably hard for basic network-complexity classes would not only uncover fundamental obstacles in the networking world (just as NP-hardness results uncover obstacles in the traditional computing world) but could also direct researchers to novel, unanticipated ways of approaching problems. Formal proofs of computational hardness have been used in cryptography; perhaps hardness of certain networked-computational problems will lead to networks in which certain undesirable behaviors (such as cheap mass mailing or denial of service) are provably infeasible.

"Future networks" means more than just "next-generation Internet," and that is the thrust of Question II.A.2. Thus, algorithmics and complexity theory should be developed, when appropriate, for all types of networks on which novel computation will take place.

> **Question III.A.1:** Formulate notions of "efficiency" for each of the computational models in Questions II.A.1 and II.A.2. Whenever possible, develop efficient algorithms for basic problems in networked computation.
>
> **Question III.A.2:** Formulate network-complexity classes and notions of "reduction" for each of the computational models in Questions II.A.1-2. Are fundamental networked-computational problems for which efficient algorithms have not been found complete for natural network-complexity classes?

Decades of successful distributing-computing research inspire confidence that study of networked computation will also be fruitful. However, the theory of networked computation is unlikely to be just a small variation on the existing theory of distributed computation; some radically new ideas will probably be needed. Even something as central as the classical ToC paradigm of an algorithm that operates on a fixed problem instance and terminates (or "converges") with a fixed, correct output after a maximum number of steps that is a function of the size of the problem instance is inadequate for

DRAFT

Internet computation, where there may not be a fixed problem instance, and computations may never terminate. For example, despite the fact that there are many "convergence" results in the interdomain-routing literature, the Border Gateway Protocol (BGP) operates not on a fixed network but on a set of Autonomous Systems, links between them, and network conditions that are always in flux; BGP never actually "converges" on a correct set of interdomain routes but rather runs continuously, adapting to announcements of newly preferred routes and withdrawals of earlier announcements. In the application layer as well, participants come and go and in distributed auctions, teleconferences, *etc.*, and the algorithms that support these activities will have to be formulated and analyzed in novel ways.

In contrast to the existing theory of distributed algorithms that provides a starting point for a theory of network algorithms, there is no clear starting point for a taxonomy of network-complexity classes and techniques for proving hardness results. One preliminary result illustrates some of the novel features of the network setting [FKSS]. A satisfactory algorithmic solution to the problem of cost sharing for IP multicast must be both computationally efficient and incentive compatible; yet, for, certain natural formulations of "computationally efficient" and "incentive compatible," it is provably impossible to achieve both criteria simultaneously (although either can be achieved if one sacrifices the other). In another context, the success of CAPTCHAs gives us reason to hope that intractability of computational problems can be put to constructive use in networked computation as it has been in cryptography; see Section III.F below.

III.B Interdomain Routing

The Internet is comprised of many separate administrative domains known as *Autonomous Systems* (ASes). Routing occurs on two levels, *intradomain* and *interdomain*, implemented by two different sets of protocols. Intradomain-routing protocols, such as OSPF, route packets within a single AS. Interdomain routing, currently handled by the Border Gateway Protocol (BGP), routes packets between ASes. Interdomain routing is currently an extremely active area of study, and it is a paradigmatic area for ToNC, because it is inherently both an algorithmic (or "protocol-design") problem and an economic (or "mechanism-design") problem.

Question II.C.1 asks whether autonomous systems (aka domains) are an inherent part of Internet architecture. Clearly, if subnetwork autonomy were to disappear, then interdomain routing as we know it would disappear as well; in particular, we would not have to contend with all of the problems that arise when subnetworks exchange traffic by following independently developed, potentially conflicting routing policies, and these are often the problems that make interdomain routing intellectually interesting and operationally difficult. Having recalled that the necessity of autonomous systems is currently an open question in the networking world, we now put that question aside for the remainder of Section III.B and assume that anything that is properly called an "internet" will need a systematic procedure for determining how to route traffic among separately administered "nets." Algorithmic and protocol aspects of interdomain routing are dealt with in this section; other aspects, including architectural issues and mathematical and economic modeling issues, are dealt with elsewhere in the report.

DRAFT

Because of the scale, heterogeneity, and subnetwork autonomy exhibited by the Internet, interdomain routes are arrived at through specification and combination of complex routing policies devised locally by individual network operators with little or no global coordination. The interaction of local policies can cause a wide range of problems in the overall routing system; some of these potential problems actually occur regularly in the real network, but others don't. Better understanding of the interaction of local policies would go a long way toward improving Internet reliability; in particular, it would be very valuable to characterize fully the situations in which potential routing anomalies actually occur and, if possible, to devise local network-operating strategies that prevent those situations from occurring. By contrast, BGP grants wide latitude in configuring local policies and has evolved over decades in response to operational needs without formal guarantees about its behavior.[2]

BGP allows adjacent nodes to exchange *update messages* that *announce* newly chosen routes and/or *withdraw* previously announced routes that are no longer available. A route announcement contains the entire path to the destination (the list of ASes in the path). BGP computes routes to every destination AS independently; so we can confine the discussion to a single destination *d*. The route-computation process is initialized when *d* announces itself to its neighbors by sending update messages. The rest of the routing tree to *d* is built recursively, as knowledge of how to reach *d* propagates through the network via subsequent update messages. Because Internet communication is *asynchronous*, update messages can be delayed. The routing process at a particular node *i* has three stages that are iteratively applied:

- **Importing routes**: Routes to *d* are received via update messages from *i's* neighbors. Node *i* has an *import policy* that specifies which of the routes it is willing to consider. All such importable routes are stored in an internal *routing table*. At any given time, *i's* internal routing table contains the latest importable routes.
- **Route selection**: If there is more than one route to *d* in the routing table, node *i* must choose one (expressing a local-policy choice among routes).
- **Exporting routes**: Whenever there is a change to *i's* best route, it announces the newly selected route to some or all of its neighbors using update messages. Node *i* has an *export policy* that determines, for each neighbor *j*, which routes it is willing to announce to *j* at any given time.

For a fixed AS graph, it is clearly desirable for the routing protocol eventually to enter a *stable* state in which every node prefers its currently chosen route to all others in its routing table, and all routing tables reflect the current route choices of its neighbors. Moreover, we would like the protocol to be *robust*, converging for every AS graph obtained by removing any set of nodes and links from the original one. It is well known that there are AS graphs and sets of routing policies for which this is not the case; furthermore, even graphs and policies that could lead to stable states may not because of the effects of asynchrony.

[2] At the Princeton ToNC workshop in February 2006, Gordon Wilfong of Bell Laboratories called the history of BGP a "horror story" that demonstrates the need for more formal, principled development of network protocols.

DRAFT

A great deal of research has been done on the analysis of BGP and the design of policies that lead to good BGP behavior. Traditionally, this work has been done in the networking community, but recently the ToC community has participated as well; indeed, successful collaboration by these two communities in the study of interdomain routing bodes well for the ToNC agenda. There is now a vast literature on the subject, containing many impressive results, but basic questions remain unresolved and constitute a central ToNC challenge.

> **Question III.B.1:** Complete the formal study of BGP. Topics of interest include but are not limited to
>
> - formal frameworks for the specification, analysis, and enforcement of policies
> - tight characterization of the conditions that guarantee robustness
> - analysis of the economic relationships (both actual and potential) among ASes and their effects on BGP behavior
> - distributed-algorithmic analysis of the time needed to reach a stable state, including analysis of the effects of asynchrony, policy changes, and node or link failures.

AS autonomy is expressed through the freedom each AS has in choosing its routes, its import policy, and its export policy. The inclusion of the entire path in route announcements allows ASes to avoid routes with loops even while making otherwise arbitrary policy choices; the entire set of routes to a given destination will thus form a confluent tree. Inclusion of the entire path is the distinguishing feature of *path-vector protocols*, of which BGP is the best-known example. Other well studied classes include *link-state protocols*, which are commonly used in intradomain routing, and *distance-vector protocols*, which are well suited for shortest-path and other metric-based routing policies.

Path-vector protocols have several advantages in the realm of interdomain routing. First, the computation is indeed performed in a distributed manner and requires communication only between neighboring ASes. Second, because the only routes considered are those announced by neighbors, and exactly one is in use at any time, the protocol enforces the requirement that routes comply with Internet *next-hop forwarding*: All routing decisions must be based solely on a packet's destination, which is accomplished if each AS has a single next hop for each destination. Third, when nodes or links join or leave the network (permanently or temporarily), these changes are announced through update messages, and nodes can quickly adapt by using alternative routes stored in their routing tables. Finally, path-vector protocols assume no prior knowledge of the AS graph; nodes learn the network topology gradually by participating in the protocol.

Although BGP and the term “path-vector protocol” have been in use for many years, formal study of this protocol class is fairly new. Algebraic methods [Sobr] and logical and combinatorial methods [GJR] have been used to formulate desirable properties of path-vector protocols, including subnetwork autonomy and policy expressiveness, and to

DRAFT

uncover inherent tradeoffs among properties. Many interesting problems in this area remain open.

Question III.B.2: Continue the formal study of path-vector protocol systems.

Alternatives to path-vector protocols may provide advantages over BGP. For example, HLP (a "hybrid link-state path-vector protocol") sacrifices some of the generality of BGP (by building some of today's AS-hierarchy structure directly into the protocol) but achieves greater efficiency by drastically reducing both the number of route updates that are necessary and the fraction of the network that can be affected by a typical update [SCE+]. Systematic exploration of such alternatives is an important ToNC challenge.

Question III.B.3: Explore alternatives to path-vector protocols. For promising alternative protocol classes, develop frameworks (of the type that are now available for path-vector routing [GJR, Sobr]) that support systematic investigation of the tradeoffs among desirable protocol properties.

A detailed and highly readable introduction to the theoretical foundations of interdomain routing (including but not limited to the topics covered in this section) can be found in [Rama].

III.C Security and Privacy in Networked Computation

The lack of security and privacy in today's networked world is one of the main reasons that "clean-slate design" of networks has become a top priority. Fortunately, security and privacy are areas in which the ToC community is ahead of the curve. There is an impressive body of theoretical work in cryptography and related areas, and the system-security world pays attention to this work. Because security of a computational system can never be demonstrated experimentally (as opposed to insecurity, which is demonstrated every time an attack succeeds), formal specifications, analysis, and proofs are valued in the security world.

The cryptography agenda in ToC has in some ways been very broad – much broader than the word "cryptography" implies, because it encompasses more than encryption and other "transformations of data." One of the ToC community's crowning achievements in the security area is a very powerful toolkit for *using data without revealing it*. Motivating applications including voting (tallying the votes and distributing the tally without revealing the individual voters' choices) and intelligence sharing (*e.g.*, computing the intersection of two terrorist-watch lists without revealing anything about the symmetric difference of these lists). In principle, general protocol constructions enable *secure, multiparty function evaluation* (SMFE [Gold, Yao]): Under weak assumptions, for any function f, any n agents $P_1, \ldots, P_n$ holding private inputs $x_1, \ldots, x_n$, respectively, can jointly compute and reveal $f(x_1, \ldots, x_n)$ while hiding all other information about the distributed data set $\{x_1, \ldots, x_n\}$. (When n is relatively small and the private data objects x_i are relatively large, SMFE is often called *privacy-preserving data mining*.) Thus, in principle, any privacy or security regime can be achieved; one needs only to supply a circuit family $\{C_n\}_{n\geq 1}$ that specifies a function family $\{f_n\}_{n\geq 1}$, and general SMFE constructions guarantee that only authorized information about $\{x_1, \ldots, x_n\}$ is revealed to the participants. Note that the sensitive data set $\{x_1, \ldots, x_n\}$ that is used without being

DRAFT

revealed resides on multiple nodes of a network, and thus the adjective "secure" describes both computation and communication

On the face of it, SMFE theory seems to be a silver bullet for privacy-preserving networked computation. It is a mature, mathematically elegant theory that considers many types of adversaries and "cheaters," each of which leads to a different (carefully formulated!) definition of "security." Nonetheless, SMFE is not in widespread use; determining why not and, if appropriate, changing this state of affairs, is a major challenge for ToNC. Clearly, one barrier to adoption is the need for special-purpose SMFE protocols that are more efficient than those provided by the existing theory, the primary goal of which has been mathematical generality rather than efficiency.

> **Question III.C.1:** Develop, deploy, and experiment with efficient SMFE protocols designed for specific applications. In particular, apply SMFE to real instances of voting, survey computation, set intersection, and order statistics *or determine why SMFE is inapplicable in these scenarios*. Explore the use of SMFE in real-world contexts that require joint computation by mutually distrustful parties, *e.g.*, intelligence, medical research, and drug discovery.

Some work in this area is already underway. For example, the FairPlay system provides a platform for specification, compilation, and execution of SMFE protocols [MNPS] that other researchers have been able to build on [FP+1]. Interestingly, attempts to deploy SMFE have revealed that protocol inefficiency is not necessarily the highest barrier to adoption [FP+2]. Technical requirements of data-mining applications, such as data cleaning and "sanity checking," are not easy to meet when data are used without being revealed. Moreover, there are non-technical barriers to adoption, including lack of clarity about whether SMFE protocols comply with laws and organizational policies that govern information disclosure. ToNC researchers can help answer these interdisciplinary questions.

> **Question III.C.2:** Clarify the barriers to adoption of SMFE and related cryptographic theory, including legal and policy barriers as well as technical barriers.

Protections should also be developed for those who use network-based information sources, including the Web. The queries a user executes may reveal non-trivial information about him or her. This issue was highlighted in August 2006, when employees of AOL released a data set including "anonymized" queries to their search engine. The idea was to allow the query set to be studied by researchers. It was quickly discovered, however, that simply removing user names was insufficient to protect privacy. One could use queries that included geographic and other related information to narrow down to a small suspect list and then examine other queries made by the same user to make a good guess at his identity. Although these query data were quickly removed by AOL from public web sites, the story demonstrates the importance of database security in the context of search engines and other interactive network-based tools. All of the major search engines, for instance, record data about their users. Even if we grant that search companies themselves should be able to use the information they gather, questions remain about how this information should be stored and protected to prevent malicious or otherwise unwarranted future use.

DRAFT

Question III.C.3: Explore the apparent tension between the good uses that search companies and researchers can make of detailed information about search behavior and the harm that can result if this information is misused. In particular, explore both technical and non-technical methods of requiring "consent" of query subjects to third-party use of data about them.

In the ToC community, the question of how one can query a database without revealing to the database controller potentially compromising information about exactly what one is looking for has spawned the field of *private information retrieval* (PIR [CGKS,Gasa]). The trivial "solution" is for the user simply to download the entire database and perform his queries locally, but this is impractical and inefficient. PIR research has produced more efficient solutions, sometimes under the assumption that there exist multiple, separate copies of the database controlled by different parties who are not allowed to communicate regarding user queries. A recent theoretical breakthrough in this area [Yekh] ties the existence of efficient PIR protocols to the existence of large Mersenne primes! In general, PIR is an existing area of study that presents both remaining theoretical-research challenges and the potential to improve security and privacy in real networked computation.

Question III.C.4: Develop, deploy, and experiment with PIR protocols. In particular, explore PIR's applicability to security and privacy in web search.

In recent years, the ToC community has taken up another central challenge in network security, namely protocol interaction and composability. Rigorous analysis of even one protocol in which adversarial behavior is anticipated can be difficult; many apparently reasonable protocols have been found faulty, and a great deal of research has been focused on methods and tools for proving correctness. In the networked world, however, proving that a protocol is secure when considered in isolation is not enough. One must consider how the protocol interacts with others. For example, there might be information leakage that can be taken advantage of elsewhere, or it may be that the protocol is secure when executed once but not when executed multiple times in parallel.

One approach to this problem that the ToC community has put forth is the Universal-Composability Framework [Can0,Can1,CKL,CLOS]. The framework supports precise specification of protocol-security requirements in a systematic way that guarantees continued security under very general notions of protocol interaction. The goal of Universal Composability is modular design and analysis of basic cryptographic ingredients that can be combined and composed freely to meet unforeseen security challenges in networked environments with decentralized control. This will remain an important goal for networked computation, and the Universal-Composability Framework may ultimately prove to be just a first step toward a complete solution.

Question III.C.5: Continue to investigate the effects of network-protocol interaction from a security point of view. In particular, continue to develop the Universal-Composability Framework.

One of the primary motivations for GENI is the need for research on "secure architectures" or "security-aware architectures": network architectures that provide primitives, abstractions, capabilities, and "set-up assumptions" (*e.g.*, secure logging, shared randomness, micropayment infrastructure, or source authentication) that are

DRAFT

provably useful for secure networked computation. ToC research in cryptography has shown that, if these primitives were available (sometimes only if they were available), we could implement SMFE, PIR, and the other algorithms and techniques in the literature [CLOS]. Networks with these capabilities could move us decisively out of the bailing-wire-and-chewing-gum situation that we are in today with respect to security and privacy. For example, we would be able to demand probabilistically checkable proofs that results of outsourced computation are correct or that code from an untrusted source satisfies certain requirements (*i.e.*, is not "malware"). More immediately, the opportunity to implement and use this broad array of techniques would meet two short-term needs: It would allow ToNC researchers to acquire operational experience with cryptographic algorithms and techniques (and thus to choose fruitful directions for future theoretical research on secure networked computation), and it would enable GENI users to create a privacy-aware network-monitoring infrastructure.

III.D Distributed Algorithmic Mechanisms

Computer Science has traditionally assumed the existence of a central planner who designs, implements, and deploys the algorithms used by computational nodes. These nodes are generally assumed either to be obedient (*i.e.*, to follow the planner's instructions to the letter) or to malfunction because they are faulty or have been subverted by attackers. This last category of *byzantine* nodes can act arbitrarily; the difficulty of modeling this type of behavior is one of the main reasons that cryptographic theory, in which assumptions are made about the computational resources available to adversaries but nothing is assumed about their motivation or utility, is so appealing.

The assumption of a central authority that designs the computational system and controls the nodes is inadequate for ToNC. The Internet has changed computation from a locally controlled endeavor to one that frequently engages many physically separate and organizationally independent people or machines or both. For example, web services, peer-to-peer systems, and even the interaction among packets on a wire are all cases in which diverse, far-flung agents find themselves jointly computing something on the Internet. Often, these agents behave *selfishly*, interested only in optimizing their own outcome. As a category of behavior, selfishness lies between the extremes of automatic obedience and byzantine disruption. Selfish actors don't deviate arbitrarily from prescribed behavior but may deviate in response to incentives, *i.e.*, the prospect of good or bad outcomes. In Economics, the field of *mechanism design* is concerned with the question of how to incentivize individual selfish agents to act in such a way that their collective actions result in a globally desirable outcome. The seminal paper of Nisan and Ronen [NR] brought the notion of computational efficiency to this study, creating the new field of *algorithmic mechanism design* (AMD) and paving the way to many interesting algorithmic results in keyword auctions, digital-goods auctions, lowest-cost routing, and other mechanism-design problems of interest.

Much of the existing work in this new field of AMD assumes the presence of a central computational facility that performs the calculations required by the economic mechanism. In eBay auctions, for example, the agents (bidders) each have independent goals, but the computation to determine winners and payments is done by the auctioneer. This combination of decentralized incentives but centralized computation applies in some

DRAFT

ToNC settings but not all. Some AMD problems are *inherently* distributed. There are many reasons that this may be the case. For example, the *data* may be inherently distributed; inputs may originate at or outputs need to be sent to a diverse and far-flung set of agents who do not know each other and cannot communicate with an auctioneer or other trusted center in the standard star pattern of classical economic-mechanism design. Second, the *computational structure* may be inherently distributed; the computation of the mechanism may be part of a larger and more complex task that must be carried out by a distributed algorithm. Third, the *social* or *organizational structure* may be inherently distributed; in the community of agents that wish to compute this mechanism, it may be infeasible to designate a single party as the trusted center. Thus, there is a need to decentralize not only incentives but also computation; this leads to Distributed Algorithmic Mechanism Design (DAMD), which was introduced by Feigenbaum, Papadimitriou, and Shenker [FPS]. Distributed algorithmic mechanisms have been developed for, *e.g.*, multicast cost sharing, peer-to-peer file management, and interdomain routing.

> **Question III.D.1:** Continue to design, analyze, and deploy distributed algorithmic mechanisms for keyword auctions, peer-to-peer file management, interdomain routing, and other core networked-computational problems.

DAMD has the same dual concerns, incentive compatibility and computational complexity, as AMD, but it differs in two important respects. First, how should one measure computational complexity? Any measure of the complexity of a distributed algorithm executed over an interconnection network T must consider the total number of messages sent over T, the maximum number of messages sent over any one link in T, the maximum size of a message, the local computation time at each node, and the storage required at each node. If a networked computation requires an excessive expenditure of any one of these resources, then its complexity is unacceptable. This is a start on the formulation of the notion of "network complexity" that has received some attention in the literature, but further research along these lines is needed.

The second major question that arises when mechanism computation is distributed is whether the interconnection network T is trusted by all of the agents. If it is, then the measure of complexity is the main difference between AMD and DAMD; in particular, incentive-compatibility concerns can be reduced (as they are in AMD and in economic-mechanism design generally) to motivating agents to reveal private information that is relevant. However, if the distributed computation is done by the agents themselves, they have more opportunities to manipulate the outcome, *e.g.*, by misrepresenting the results of a local computation to a neighboring agent or, more drastically, by simply not communicating with that neighboring agent at all, in an attempt to exclude him from the game. Thus, distributed algorithmic mechanisms must provide incentives to guarantee that selfish agents find it in their best interest to perform the distributed computation as specified.

The need to incentivize computational agents to follow protocols as well as to reveal information truthfully has led DAMD researchers to nontraditional "solution concepts" or notions of equilibrium. Traditional solution concepts such as dominant-strategy equilibrium and Nash equilibrium are either too strong or weak, and they are designed primarily to incentivize truthful revelation of preferences. In DAMD, the notion of *ex-*

DRAFT

post Nash equilibrium has received a great deal of attention. Informally speaking, in ex-post Nash equilibrium, an agent will self-interestedly follow the specified algorithm if all other agents are doing so, regardless of the other agents' private inputs to the algorithm. In network-protocol terms, ex-post Nash equilibrium is a useful concept when networked-computational agents all use "out-of-the-box" protocol software but insist on keeping their protocol-configuration parameters private. Although weaker than dominant-strategy equilibrium, ex-post Nash equilibrium is a fairly strong solution concept; it does not require strategic agents to have any knowledge of or to make any assumptions about the private system-configuration information of other agents. Contrast this with the standard Nash-equilibrium concept, in which agents are assumed to know the inputs ("moves," in game-theoretic terms) of other agents; in the interdomain-routing context, this would mean that ASes are assumed to know the local routing policies of other ASes, which is unrealistic.

These notions of network complexity and equilibrium are first steps toward the desiderata for networked computation, but considerably more work is needed.

> **Question III.D.2:** Develop a comprehensive and rigorously formulated set of properties that distributed algorithmic mechanisms should satisfy. In particular, further explore the notions of complexity and equilibrium that have been used in the existing DAMD literature, and develop new notions as needed.

In Section III.C, we pointed out that protocol interaction is an important issue in the struggle for secure networked computation. Similarly, mechanism interaction is important in the struggle for incentive-compatible networked computation. Internet-based payment mechanisms, such as PayPal, are widely used and may be subject to manipulation of various sorts; how should they be integrated with other distributed mechanisms so that incentives and fraud-control measures developed independently by various mechanism designers do not interact in unfavorable ways? In web-based commerce, an issue for sellers of advertising space such as Google and Yahoo is that their return is based not only on the bid of the advertiser, but also on the probability that the ad is clicked by the end user [MSVV]. In general, agents in distributed-mechanism computations are likely to engage in collusion, approximation, adversarial behavior, and other things that are not modeled adequately by existing DAMD theory. The theory should be expanded to accommodate these realistic scenarios.

> **Question III.D.3:** Explore the roles of mechanism interaction, approximation, collusion, and adversarial behavior in distributed algorithmic mechanisms and in definitions of equilibria.

III.E Coding and Communication

For many years, information theory was not a major focus of the ToC community; it was studied primarily in engineering departments. Over the last decade, inspired by newly discovered connections to algorithms and complexity theory, ToC researchers have invested significant energy in information-theoretic problems. For example, major

DRAFT

advances in low-density parity-check codes [LM+1, LM+2] and list decoding [Suda, GS] have been made in Computer Science.

During this time, information theory has moved from the sender-channel-receiver model, in which the transmission medium is viewed largely as a black box, toward a fuller, more computational model of the underlying network. ToC, with its long history of work on distributed computation, has a great deal to offer in this research effort. Two representative problem areas in which ToC can contribute are network coding and the information-theoretic foundations of mobility.

In standard store-and-forward networks that use coding, encoding is done at the source, and decoding is done at the destination. Network coding considers the possibility that intermediaries in the network have sufficient power and intelligence to do more than just store and forward packets. Instead, they can actively combine or otherwise transform packets they have received to create new pieces of information for transmission. What are the implications of this change in structure?

As a simple example of how network coding can lead to improvements, consider a wireless or sensor network in which two agents X and Y communicate through a relay R. Agents X and Y can send to R but not to each other; the relay R can send to both X and Y and can also broadcast a message to both of them, as they are both in range. Agent X wishes to send the message x to Y, and Y wishes to send the message y to X. We assume that the messages are a single packet in length. The natural way to do this uses four transmissions: X sends x to R and R forwards it to Y, and similarly in the other direction. Using network coding, it can be done with three transmissions as follows: Agents X and Y send their messages to R, and R then broadcasts the exclusive-or of x and y to both. With the exclusive-or, both X and Y can recover the desired information; the end result is the same, but the communication cost is lower.

Network coding has received a tremendous amount of attention in the last few years, but many basic questions remain open. They provide an interesting theme for ToNC by positing that computation should be a fundamental part of communication, while the point of departure for most of the questions in this report is that communication should be a fundamental part of computation. The groundbreaking paper of Ahlswede, Cai, Li, and Yeung [ACLY] established the area of network coding in 2000, demonstrating a fundamental connection between network information flow and the max-flow-min-cut theorem. Further connections have been made to multicommodity flow [AHJ+] and Steiner-tree problems [AC]. Indeed, perhaps the key open problem in network coding is algorithmic: Can decoding for network codes be made faster algorithmically, perhaps in a manner similar to that used in low-density parity-check coding [LM+1, LM+2]? We expect network coding to flourish in coming years, fortifying the connections between the information-theory and ToC communities.

> **Question III.E.1:** What are the benefits and costs of network coding in realistic network structures? What tradeoffs can be obtained among computation, latency, buffer size, and other network resources when using network coding? Are there more efficient algorithmic methods for network decoding?

DRAFT

In an extreme version of network coding, known as the *black-box channel* model, almost all of the "computation" part of "networked computation" is subsumed by the communication channels. In this model, channels are described by properties ("bit-hiding channels," "anonymous channels," "authenticated channels," *etc.*); the composition of two channels is a channel, and thus network algorithms and protocols that are themselves channels can be built by composing channels. It may be possible to leverage known reductions among properties (including, but not limited to, security properties) to prove both upper and lower bounds on the complexity of protocol-design tasks and to develop a useful notion of universality, as discussed above in Section II.A.

Closely related to network coding is the study of information transmission over mobile networks and, in particular, over mobile ad-hoc networks (MANETs). The information-theoretic point of view focuses on the achievable rate at which information can flow through the network, often in terms of specific source and destination pairs, and on means for achieving this rate in practice. However, network performance is also inherently tied to the underlying geometry of the network; geometric considerations have not received a lot of attention in the engineering community's study of MANETs, but they are an active area of study in ToC. Random-graph theory asks questions such as "what is the minimum edge density at which a network is likely to be connected?," or "how one can effectively construct matchings on random graphs?" Such questions can play a fundamental role in understanding capacity limitations of MANETs.

> **Question III.E.2:** Develop formal models of MANETs. Explore the use of geometry and random-graph theory to capture the total communication capacity of MANETs.

The study of MANETs presents algorithmic challenges as well as mathematical-modeling and information-theoretic challenges. For example, both incentive compatibility and security are poorly understood in the MANET context. Ad-hoc networks in general, and MANETs in particular, are often short-lived; thus, subscriptions and other long-lived financial and management arrangements may not be applicable. As explained in Question II.C.6, network resources themselves may serve as currency when monetary transfers aren't applicable (as bandwidth does in BitTorrent), but exactly which resources can serve in this way is not clear. Similarly, public-key infrastructure and other security arrangements that rely on long-lived relationships aren't easy to deploy in ad-hoc networks. In general, MANET computation is an even more wide-open area than MANET engineering.

> **Question III.E.3:** Design and implement algorithms that can be executed reliably on MANETs. In particular, develop mechanisms that ensure security and incentive-compatibility in short-lived network environments.

Network coding and MANETs are just two among many areas in which theoretical Computer Science and theoretical EE can work together on the ToNC agenda. Joint work by these two communities may benefit academic culture as well as ToNC research.

DRAFT

III.F Search: Networks as Information Sources

Search engines have greatly enhanced the utility of the World Wide Web. Who could have imagined a decade ago that the web would grow to its current size 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. Enhancing the power of search is a central element of the ToNC agenda.

Security and privacy challenges in search are touched upon in Section III.C above, but there are many other challenges as well. For example, almost all practical implementations of search today are keyword-based. This greatly limits the ability to find data that is not text-based and, in particular, to find images, video, audio, or database records, even if they are on public websites and in standard formats. There is every reason to believe that the ToC community can make progress in this area; it has played a key role in the development of ranking algorithms now used in search engines, most notably in the formulation of the HITS algorithm [Kle0] and the PageRank algorithm [PBMW]. Although they are used today to answer keyword searches, the breakthrough contribution of these algorithms was not better text analysis but rather a systematic way to exploit the link structure of the web.

> **Question III.F.1:** 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 supports keyword search.

One research area that may greatly improve search but has only recently received attention in ToC is human-aided computing. The most successful outcome of collaboration between these two fields has been the creation of CAPTCHAs (Completely Automated Public Turing Tests to Tell Computers and Humans Apart [ABL,N]), which are tests that distinguish humans (who are the intended users of web-based services) from computers (which can be programmed to abuse these services), by posing problems that are apparently hard for computers but easy for humans. CAPTCHAs are in widespread use today. In the tradition of public-key cryptography, computational intractability has been turned into an asset instead of a liability. 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

DRAFT

different things when applied to humans from what they mean when applied to computers.

> **Question III.F.2:** Develop the theoretical foundations of human-aided networked computation; in particular, develop algorithms that allow networked computers to leverage and aggregate the results of millions of human actions. Explore the power and limitations of increasing human involvement in network-based search.

IV. Solving Problems that are Created or Exacerbated by Networks

In addition to new network designs and new computational models, the ToNC-research agenda includes algorithmic questions in established models of computation (*e.g.*, streaming, sampling, and sketching algorithms to answer questions about massive, network-generated data sets) and some purely mathematical questions (*e.g.*, generative models for various sorts of "network-like" random graphs). Interestingly, some of the challenges raised by pervasive networked computation are not primarily technical in nature; they are as much questions about philosophy, law, and social norms as they are about mathematics, algorithms, and technology. This is particularly true in such areas as data privacy, copyright, and electronic commerce. We explore these aspects of ToNC in this section.

IV.A Privacy and Control of Sensitive Information

Much of 20th-century computational theory equates "privacy" with confidentiality or secrecy. This approach has proven to be inadequate for networked computation, in which more and more sensitive information about people and organizations is created, captured, and stored by the computers and networks that mediate our daily lives.

It is our thesis that draconian use of encryption and access control, even if its development and deployment were technologically feasible, would not create a world in which legitimate work proceeds unimpeded but sensitive information is never misused. Networks are increasingly popular precisely *because* they enable people and organizations to share far more information than ever before. Is there any intuitive notion of "privacy" that is consistent with that basic fact? Rather than *hiding* sensitive information entirely from all but a small number of people or machines that are identified before the information is created (the traditional goal of the cryptographic-research world), can some combination of technology, law, organizational policy, and social norms ensure *appropriate use* of information by the dynamic and potentially large set of people and machines that may have legitimate access to it over the course of its lifetime?

The ToNC community can contribute by formulating and exploring "less black-and-white" privacy frameworks that capture stakeholders' requirements while allowing for realistic solutions to problems. For example, there are existing cryptographic techniques, such as anonymous electronic payments [Chau], that would enable transactions between

DRAFT

businesses and customers without requiring the exchange of personally identifying information (PII). These techniques have not been widely adopted, at least in part because *businesses do not want to give up this information*. As discussed in Section III.C, the abstract goal of "using data without revealing it," which has been promoted for almost 25 years by cryptography researchers in the ToC community, fails to model some practically important aspects of real-world data-processing problems, *e.g.*, data cleaning. This leads to the following general question about models and definitions.

> **Question IV.A.1:** Is there a reasonable, enforceable notion of privacy that allows businesses to collect and store PII about their customers?

In their pursuit of privacy-preserving computation, cryptographers in the ToC community have concentrated on interactive protocols, *e.g.*, SMFE and PIR; these lines of research lead to interesting problems in "computing on networks," some of which are discussed in Section III. However, in today's world of massive databases and the potential (wrought by high-bandwidth networks) to publish them or to share them selectively with partially trusted collaborators, it is desirable to consider non-interactive techniques as well. Two complementary scenarios in which ToC cryptographers have explored broader and potentially more useful notions of privacy are "privacy in public databases" [CDM+], in which the goal is to modify a database so as to support aggregate queries but protect individual data subjects (think census), and "group privacy" [NS], in which the goal is to modify a database so as to support retrieval of individual (or small, precisely specified sets of) records but prevent retrieval of large or imprecisely specified sets of records (think airline-passenger databases). Some positive results have been obtained (*e.g.*, it is possible to encode a database of names and email addresses so that one can look up the email address of someone whose name one knows but cannot "mass harvest" all of the email addresses for spamming purposes), but there is not yet a complete characterization of the sets of queries that can be enabled precisely.

> **Question IV.A.2:** Explore database encodings that (provably) permit the (efficient) retrieval of some sensitive data and prevent the retrieval of others. In particular, consider the use of "gray areas," in which some properties of the original database are readily revealed by the encoded version, some are provably hidden, and no guarantees can be made about others.

Complete solutions to these problems will have to incorporate essential aspects of networked computation, *e.g.*, the massive scale of modern databases, the interplay of cryptography and coding theory, and the fact that interested parties may try to lie about their identities in an attempt to gain unauthorized access to sensitive information.

Several reasons that SMFE, PIR, and other ToC formulations of secure networked computation, while technically elegant and perhaps somewhat useful, may not be sufficient have already been given, but there is at least one more important reason: These constructions (and cryptographic theory in general) have nothing to say about which properties of sensitive data sets should be revealed and which should be hidden. How can one assess the consequences of revealing a particular functional value $f(x_1, \ldots, x_n)$ of a sensitive data set $\{x_1, \ldots, x_n\}$? What information might relevant parties already possess that could be combined with $f(x_1, \ldots, x_n)$ to affect the participants in nonobvious ways?

DRAFT

> **Question IV.A.3:** Develop formal methods for determining which properties of sensitive data sets "should" be revealed in various contexts. In particular, develop formal methods for assessing the prospects that relevant parties could combine revealed information with pre-existing knowledge in harmful ways.

A start on multidisciplinary consideration of what information "should" be revealed in various contexts can be found in Nissenbaum's work on *contextual integrity* [Niss]. Note that the prospect of an organization's publishing data that it has determined to be harmless (perhaps after an attempt at sanitization) and then learning that people have combined that data with pre-existing knowledge in harmful ways is not at all hypothetical: This is precisely what AOL was vilified for in August 2006.

Finally, recall that sensitive data need not be "private." Rather, they must be handled in accordance with agreed-upon rules, if none of the stakeholders is to be harmed. Copyright is the canonical area in which use of data is clearly rule-governed, but the traditional notion of privacy as confidentiality is useless: Published, copyrighted works are the sensitive data objects in question, and they generally are not supposed to be kept confidential; indeed, many of them are supposed to circulate widely. Just as increasing use of powerful, networked computers in the course of everyday business transactions necessitates a rethinking of "privacy," increasing use of powerful, networked computers in entertainment and other creative work necessitates a rethinking of copyright. Existing copyright law is in some ways ill-suited to the regulation of digital works, *e.g.*, because it depends heavily upon the copyright owner's right to control copying (an essential operation in *every* use of a digital work, not just those that could conceivably be construed as infringement) and because it specifies owners' rights much more clearly than it specifies users' rights (thus leaving open the possibility that digital-distribution regimes that are far more restrictive than traditional analog-distribution regimes could effectively moot fair use and other types of access that users have long enjoyed but that, strictly speaking, are "defenses" against charges of infringement, rather than affirmatively specified "rights"). On the other hand, the fact that digital works can be copied, distributed, and modified far more easily than analog works threatens to undermine some legitimate owners' rights. Is there a way simultaneously to fulfill the US Constitutional vision[3] of "promot[ing] progress in science and the useful arts" through intellectual-property rights and to exploit the vastly increased power that computers and networks bestow upon creators and users?

> **Question IV.A.4:** Explore the co-design of legal systems and technological systems that support the creation, distribution, and use of digital copyright works.

For an in-depth discussion of the interplay of law and technology in the copyright arena, see, *e.g.*, [CIPR].

IV.B Incentives and Economic Models

In formulating new definitions and computational models for the networked environment, it is extremely important to take economic and incentive considerations into account. For

[3] *Article 1, Section 8, Clause 8*

DRAFT

example, most definitions, models, and metrics in established cryptography and security theory focus on providing worst-case guarantees, typically in a stylized model of a network of interacting agents and typically from the viewpoint of a single agent. It is now necessary to approach security at the network level rather than at the agent level and to provide quantitative measures of security with respect to realistic models of user behavior rather than absolute guarantees of security with respect to a stylized model of behavior. Useful metrics should permit comparison of the cost to deploy security measures with the expected benefit to the system. Following seminal work by Anderson [Ande], researchers are now using economic theory to go beyond simply proving that a technology is secure and also establish that, with the proper incentives, it will actually be deployed and used. This economic approach may, for example, guide the development of general techniques for comparing locally deployable security technology (*e.g.*, client-side spam filtering), for which individual users bear the responsibility and over which they exercise control, to centrally deployable security technology (*e.g.*, server-side spam filtering), which users do not have to take responsibility for but also cannot control.

In formulating appropriate security models for networked computation, one must keep in mind that security *per se* is rarely, if ever, the primary goal of users; instead, users typically want to accomplish a specific goal (*e.g.*, web searching or email), for which they use a specific network service or protocol and expect a certain level of performance at a certain cost. A plausible definition of a "secure" service or protocol is one that maintains good performance in the presence of adversaries. With such definitions in hand, it is natural to ask the following type of question:

> **Question IV.B.1:** When, if ever, does security raise the cost of network services and protocols? If secure services and protocols are indeed more expensive to design, build, and use, how much more expensive are they, and are users willing to pay this increased cost?

Privacy (of users, service providers, product vendors, and all other stakeholders) is another aspect of networked computation that cries out for economically informed definitions and models. In practice, users of network-based services are often willing to give up privacy, *e.g.*, by revealing PII to e-commerce service providers, presumably because they value something that they receive in return. However, this tradeoff has not been successfully formulated or quantified. Similarly, the cost of data discovery plays an important role in the *perception* of privacy but is not modeled by current theories. For example, the availability of conventional phonebooks (keyed on people's names) and the availability of reverse-number-lookup phonebooks (keyed on phone numbers) are perceived very differently from a privacy point of view, in spite of the fact that they contain exactly the same information. On a more basic level, the migration of "public records" from paper to websites has changed the practical, if not the legal, meaning of "public": The ability to learn the Social Security Number of a person whose name one knows by going to a court house and retrieving a property deed is much less valuable than the ability to find the same deed on a local-government website. This discussion leads to the conclusion that ease of information access should play a role in a theory of privacy.

> **Question IV.B.2:** Develop an integrated theory of privacy, economics, and networked computing. In particular, formulate and quantify the relationship

DRAFT

between the computational cost of data discovery and the practical privacy of data subjects and data owners.

Economic and incentive issues enter the theory of networked computation on many levels. For example, the phrase "clean-slate design" has become a mantra in networking-research forums and in calls for proposals. Not surprisingly, many people have raised the question of whether anything that requires a "clean slate" could ever be brought to fruition in a world in which networked computation is pervasive and mission-critical. From a research perspective, one crucial point is that clean-slate *design* does not presume clean-slate *deployment*. Part of the ToNC agenda is the evaluation of new technologies, methods, algorithms, *etc.* from the perspective of incremental deployability and paths to adoption.

Question IV.B.3: Develop techniques with which to assess proposed networked-computational systems with respect to adoptability, deployability, and migration paths from current technology. Identify good test cases with which to assess progress on adoptability, particularly of security technology.

Unquestionably, the current Internet-protocol framework is problematic. Protocol behavior, especially cross-protocol interaction, is poorly understood. Network operation, management, and troubleshooting are horrendously difficult. Operational goals and policies are difficult to express, and small changes in policy can lead to big, unpredictable changes in behavior. This unsatisfactory state of affairs is one major motivation for the clean-slate design effort, and many of the research questions discussed in Sections II and III above grow out of that effort.

Nonetheless, critical examination of the need for clean-slate design of a next-generation Internet may also fruitful. Although we have known for years of the existence of operator choices and user behavior that could cause total chaos and network meltdown, this worst-case scenario has not materialized, and the Internet has continued to grow more useful and more central to everyday life. Can we formalize, both descriptively and prescriptively, the reasons that worst-case scenarios have not materialized? The work of Gao and Rexford [GR] provides an excellent example of this type of analysis. The basic question addressed therein is why the BGP interdomain-routing system works as well as it does in practice, in view of the fact that, in theory, ASes may choose policies that lead to BGP divergence or oscillation – states in which users in different Internet domains may be unable to communicate with each other. Gao and Rexford formulated three conditions on the routing preferences of and commercial relationships among ASes that guarantee BGP convergence and stability; moreover, they observed that these conditions accurately describe the current structure of the commercial Internet and that border routers can, in practice, be configured to conform to them. More results of this type might provide an alternative to clean-slate design.

Question IV.B.4: Develop methods for the identification of operating regimes and market structures in which particular network protocols work well, despite the existence of conditions in which these protocols are known to fail or to work poorly.

Similar questions can be investigated at a higher level of abstraction than that of specific network protocols. Any long-lived, large-scale networked system can be viewed as an

DRAFT

equilibrium, even though it may have grown and cohered in an ad-hoc manner, rather than as a result of an explicitly formulated game. If we could formulate games of which existing successful networks are equilibria, we might be able to predict whether proposed changes to these games would result in more desirable equilibria.

> **Question IV.B.5:** Develop methods for "reverse engineering" networks to discover the social processes and incentive structures that created them.

Finally, we remark that explicit use of market mechanisms in networked computation may result in the integration of network activity with existing and future financial markets and thus in the importation to the Internet of financial-market risks and volatility. For example, consider the use of bandwidth and other network resources as currency, as contemplated in Question II.C.6. If bandwidth, say, were to become a widely traded commodity, might this lead to speculation and other "purely financial behavior" that is entirely unmotivated by network performance? Would such markets lead to more efficient allocation of network resources than those arrived at by network operators, or would they drastically increase the incentive for denial-of-service attacks and other disruptions by legitimizing the financial rewards for such activities? Or both? The general observation is that any financial-incentive mechanism that is widely adopted in cyberspace may interact with the wider financial system and that algorithmic-mechanism designers should take this into account.

> **Question IV.B.6:** Determine how one can leverage finance, insurance, and other established market structures in the course of developing mechanisms and incentive systems for networked-computational activities – instead of being thwarted by incompatibility with these established markets.

IV.C Analytical Paradigms

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 the context of the particular research project that they are part of.
- **Model**: Propose an underlying model for the observed behavior.

DRAFT

- **Validate**: Find data to validate (and, if necessary, specialize or modify) the model.
- **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 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.*).[4] 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.

> **Question IV.C.1:** 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 exactly what constitutes convincing validation. Fleshing out this area is a basic ToNC challenge.

> **Question IV.C.2:** 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.

> **Question IV.C.3:** 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.

---

[4] *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.

Answering Questions IV.C.1-3 will involve a mix of theoretical and experimental work and will obviously require access to data, some of which may have to be acquired through sophisticated network monitoring. GENI should provide an opportunity to gather the necessary data.

There are also purely theoretical problems that beckon in the area of analytical paradigms for networked computation. For example, holistic network models remain elusive. Previous work has focused on one observable at a time, *e.g.*, network formation, network traffic, or a particular network interface to the external world. Models that describe and predict many features of a networked system would be useful, particularly if they could be tuned to accommodate different applications that have different requirements. In the area of mathematical methods, the network analog of smoothed analysis would clearly be useful [ST]. In standard (one-machine) computational complexity, smoothed analysis is an alternative to the extremes of worst-case running time (as a function of n, the maximum, over all x of length n, of the running time on input x) and average-case running time (as a function of n, the average running time over R, where R is a distribution on instances of length n). The smoothed complexity is the maximum, over x of length n, of the average over R of the running time on instance $x + \varepsilon R$. 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.

> **Question IV.C.4:** 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.

Interesting mathematical problems can also be found in the area of Erdos-Renyi style generative network models. Existing work has focused on the generation of random graphs with specified degree properties (*e.g.*, [FKP]); many challenges remain in the generation of graphs with specified conductance or spectral properties [MPS].

Finally, new analytical paradigms may help us capture some of the social and political aspects of networked computation. In today's peer-produced information environment, anyone can contribute "information," but quality-control mechanisms vary widely across information-retrieval systems and are sometimes entirely absent. Is "democratization" inherently at odds with accuracy, or can "social-certification" mechanisms be developed to improve information quality? Also on the border of mathematics and social science is the study of "small-world phenomena" in networks. Much is known about how to exploit small-world structure for routing information through networks; less well understood is the influence of small-world structure on the diffusion of behavior in networks, including fads, opinions, and adoption of new products [Kle1].

DRAFT

IV.D Massive-Data-Set Algorithmics

We have already remarked several times that 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 and that, as a result, fresh thinking is needed about data privacy. Here, we point out that this situation also strains 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 ToC community has already taken up this challenge on multiple fronts. New computational models have been developed, including data streaming [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 ToC 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; representative challenges in search are presented in Section III above. In this section, our goal is simply to observe both that network elements (such as routers and web servers) routinely generate potentially interesting massive data sets, that powerful networks give users massively increased access to potentially interesting data of all sorts, and that algorithmic challenges abound.

> **Question IV.D.1:** Continue the development of MDS algorithmics. In particular, develop
>
> - additional lower-bound techniques in the streaming, sampling, and sketching models,
> - MDS algorithms that can handle strategic or adversarial data sources,
> - MDS algorithms for complex data formats, including images, video, and audio.

If the possibility of strategic or adversarial behavior by data sources seems far-fetched, note that, in fact, it is already in evidence: Web searching is a prime example of MDS computation, and, as remarked in Section III, Web-site owners regularly exhibit strategic and adversarial behavior.

Massive-graph algorithms are an important part of the ToNC agenda; they have already received some attention (*e.g.*, [FKM+]), but many challenging problems remain open. The following two questions exemplify the discussion of massive-graph problems at the ToNC workshops.

DRAFT

Question IV.D.2: 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?

Question IV.D.3: If a (single) computer (that is not a node in the evolving graph under consideration) can compute or approximate these values, can it also efficiently prescribe corrective action when problems are detected?

IV.E Specification and Verification

Specification and verification of hardware components and software programs is an established area of Computer Science; in the networking area, there has been some success in specifying and verifying individual protocols (*e.g.*, cryptographic protocols [Mead]). However, only recently have specification and verification techniques been applied to network services and the influence of packet transformations that they perform. As a specific example, consider firewalls, which are ubiquitous and indispensable defense mechanisms used in business and enterprise networks. Just as router misconfigurations can lead to unpredictable routing problems, misconfigured firewalls may fail to enforce the intended security policies or may incur high packet-processing delay. As distributed firewall rules are concatenated, it becomes extremely difficult to predict the resulting end-to-end behavior and to decide whether it meets the higher-level security policy. Similar issues arise in other settings in which packet transformations or exclusions may occur, including packet filtering and quality-of-service mapping.

Symbolic model checking has been used successfully to find security and reliability bugs in large programs; the checker examines the control-flow and/or the data-flow to determine whether a program satisfies user-specified properties without running the program. The concatenation of the configuration rules of firewalls can be viewed as a specialized software program. After classifying all possible policy anomalies (including both inconsistency and inefficiency), firewalls can be modeled as finite-state transition systems. Symbolic model checking techniques can then be applied to these finite-state representations to detect violations and inconsistencies at different levels: intra-firewall, inter-firewall, and cross-path [YMS+].

Instead of finding bugs, one may want to guarantee a certain quality of service. In a typical firewall setting, packets are compared against a list of rules sequentially until the packet matches a rule. The total number of rules configured and the order in which they are applied play major roles in firewall efficiency. Firewalls with complex rule sets can cause significant delays of network traffic and become attractive targets for DoS attacks. To optimize packet filtering and provide network Quality of Service (QoS) guarantees, it would be desirable to apply verification-based techniques, proactively preventing vulnerabilities in firewalls by using static analysis before actual deployment.

Question IV.E.1: Explore the application of specification and verification techniques to network services and structures. Specifically, apply established techniques from device and program verification to the analysis of end-to-end

DRAFT

properties of computer networks, or develop new validation techniques as needed.

Validation of end-to-end properties can be done by measurement and monitoring infrastructure as well as by formal analysis tools. Much of the thought about measurement and monitoring, however, has focused on optimization, *e.g.*, how to measure latency in order to optimize routing? (See Section II.E.) Monitoring schemes have also been suggested for purposes of accounting, *e.g.*, for finding the elephants (or large flows) passing through a router.

Suppose instead that we design a measurement and monitoring infrastructure to detect or potentially anomalous network behavior before it becomes problematic. Such a monitoring system might detect a small number of malfunctioning components and isolate them before the problem spreads to the rest of the network; it might allow forensic examination of evidence for the purpose of identifying attackers and improving defenses. Much of this type of monitoring might be done on a component basis, instead of a flow or connection basis. In effect, we would like the network to perform basic self check-ups, with the goals of stopping the equivalent of individual colds before they become epidemics and of identifying outbreaks of new, more virulent problems. This process would complement formal verification methods; it will be used to diagnose problems in deployed systems and issue warnings, rather than to prevent deployment of vulnerable systems.

Question IV.E.2: Develop tools that perform routine network check-ups and take action to prevent problems from spreading. How can data collected from such tools be organized and stored in a manner that facilitates forensic analysis?

Ongoing work on analysis of network-security data that includes a ToC component can be found in, *e.g.*, [CTA].

IV.F Design and Implementation of Wireless and Sensor Networks

New technologies and infrastructures have led to fundamentally new types of networks, including wireless and sensor networks.[5] These networks have different characteristics from those of the wired Internet and therefore motivate a wealth of interesting open problems. There has been an explosion of research in this area, and we cannot possibly cover it all; our focus in this section will be on three research areas in which ToC plays a particularly active role: scheduling, routing, and algorithms for constrained devices.

Scheduling

[5] The definitions of these types of networks are themselves somewhat fuzzy. We use the term "wireless network" to refer to networks that primarily serve mobile users; these users may be communicating with fixed-location base stations, mobile base stations, or other mobile users, depending on the context. By contrast, we take as the defining characteristic of "sensor networks" the fact that nodes have relatively little communication and computational power. Of course, one could have wireless sensor networks, and wireless technology can also provide communication among non-mobile nodes.

DRAFT

In networks in which mobility is possible, a key scheduling question is whether to serve a nearby user at a higher rate or a distant user at a lower rate. Optimization problems with multiple users and transmitters are challenging even in static settings; when the users' positions (and hence the available service rates) change dynamically, the optimization process is even more challenging.

Much of the previous scheduling work in the networking area deals with average-case, stochastic models, where service rates and requirements are given by stationary stochastic processes. It is unclear that these models are suitable in a mobile, wireless setting; stationarity may be destroyed unless strong, largely unjustifiable assumptions are made about the nature of the mobility. ToC, with its historical emphasis on worst-case algorithmic analysis, has much to offer this line of research. Although systems will not be optimized for the worst case, it may be important to know what the worst-case behavior is, because it gives insight into how the system might break down, *e.g.*, an emergency or disaster scenarios. Alternatively, one might design a system that gives up some performance in the average case in order to improve the worst case.

> **Question IV.F.1:** What insights does worst-case analysis offer for scheduling strategies in wireless networks? Can one design systems that offer near-optimal average-case behavior while maintaining reasonable worst-case performance guarantees?

Routing

Wireless and sensor networks have rich geometric structures. Determining users' current positions and taking advantage of these locations can improve basic network functions, including routing. Most existing geometric-routing schemes are based on moving closer step by step to the final destination. But greedy forwarding runs the risk of not taking the optimal path or, more extremely, of reaching a dead end. Some progress has been made in this area [KK], but existing solutions are not optimal; in particular, how best to deal with obstacles to communication (including both fixed obstacles like walls and buildings and intermittent sources of interference) is an open problem in geometric routing.

> **Question IV.F.2:** How can geometry best be exploited in wireless networks? What mechanisms can be used to deal with interference of various types? What are the alternatives to greedy forwarding and, more generally, to geometric routing?

The question of where to put fixed or moving resources (including routers and data sinks) within a wireless or sensor network also presents algorithmic challenges. The underlying geometric structures of these networks suggests that several well-studied ToC problems, including the Steiner-tree and the facility-location problems, may have something to offer. For example, the question of where to place a limited number of data-collection units (sinks) so as to minimize the cost of sending sensor data to a central hub can be seen either as a Steiner-tree or a facility-location problem, depending on the forwarding capability of the sensors. Algorithm advances on basic geometric problems may thus pay off immediately in wireless and sensor networking.

DRAFT

Question IV.F.3: Explore the application of classical geometric algorithms, including Steiner-tree and facility-location algorithms, to routing and other problems in sensor and wireless networks. Consider new variations on these problems that capture sensor- and wireless-network requirements.

Interestingly, advances in geometric routing may find subsequent use in the wired Internet. In overlay and peer-to-peer networks, nodes can be assigned "network coordinates" in a multi-dimensional geometry in order to capture significant features of network performance, such as round-trip time [CDK+, DCK+]. These coordinates need not be static; as network properties such as latency or bandwidth change, the network coordinates can change accordingly. In this sense, the wired Internet can be modeled as a mobile network inside some Euclidean (or non-Euclidean!) space, and algorithms or methodologies developed for wireless or sensor networks can be used if they provide significant benefits [LPMS, PLMS]. Work by Kleinberg, Slivkins, and Wexler [KSW] provides an initial theoretical framework for this approach, showing that low-dimensional embeddings that are accurate for most distances can be achieved with very limited infrastructure.

Question IV.F.4: Explore the applicability of geometric routing in wired networks. What are the theoretical limits of embeddings into small-dimensional geometric spaces, and what do they imply about routing? Is there a natural role for non-Euclidean geometries in this setting, and how does one route naturally in non-Euclidean metrics?

Question IV.F.5: When using geometry in wireless and sensor networks, consider embeddings into higher-dimensional geometries in order to incorporate additional information about the nodes that cannot be incorporated in low-dimensional embeddings.

Constrained Devices

Current sensors (or variants such as RFID tags) have very limited communication and computational capabilities. Although these capabilities are likely to improve in the long run, the push for very small and low-cost sensors is likely to leave us with heavily resource-constrained nodes for the foreseeable future. In particular, for devices with limited battery life, communication can be extremely expensive; even the act of waking up to see whether a message is arriving can be significantly more expensive than a large chunk of computation. Memory capacity is also a potentially significant constraint.

In such a setting, many distributed computing problems that have been solved in the theoretical literature, such as consensus, leader election, clock synchronization, and even broadcast, become interesting again, because the change in the underlying network model renders existing solutions unusable. Problems that are more specific to sensor networks, such as data aggregation, also force us to rethink basic distributed algorithms and, in particular, to rethink the tradeoffs between communication and computation. Finally, security challenges abound in this new environment, including cryptography that can easily be implemented on constrained computing devices and protocols that work smoothly even when "listening isn't free."

DRAFT

Question IV.F.6: Rethink "solved" problems in the distributed-algorithms and security-protocol areas in light of the prevalence of networks with severely resource-constrained nodes.

V. Conclusions and Discussion

From the presentations and discussions at the ToNC workshops held in Spring 2006, we have culled a very broad set of research questions that we hope will engage a large fraction of the ToC community. This set is not meant to be an exhaustive list but rather a large enough sample to demonstrate conclusively that there is a substantial role for ToC researchers to play in the design, development, analysis, operation and use of future networks. We conclude this report with a brief discussion of challenges and potential obstacles to progress on this agenda.

Institutional Support of ToNC

The ToC community can pursue the ToNC-research agenda on many fronts and in many ways. Valuable types of research projects include but are not limited to:

- **Small, single-investigator, purely theoretical projects:** By "small," we mean funded at a level sufficient to pay for one or two months' of PI summer salary per year, one or two PhD students per year, and a few incidentals such as conference travel or commodity computers for the project participants.
- **Medium- and large-sized, multi-investigator projects involving both theory and experimentation:** The distinguishing features of such a project are (1) multiple PIs, at least one of whom is a theorist and at least one of whom is an experimentalist and (2) the inclusion of experimental work on a "real problem" arising in a network that can be built or at least envisioned in the current technological environment. Funding levels for these projects can range from anything that is bigger than "small" up to several million dollars per year.

Program Directors in NSF's Computer and Network Systems Division have explicitly welcomed the type of medium- and large-sized project proposal described here, and the "distinguishing feature" text above comes from them. Careful consideration was given at the workshops to whether small, purely theoretical projects are equally important for success of the ToNC agenda, and participants decided that they are, for two basic reasons: (1) The intellectual scope of ToNC should not be limited by networks that can be built or even envisioned in the current technological environment; technologically untethered but mathematically rigorous investigation of networked computation is also worthwhile. (2) Some of the most eminent and productive members of the ToC community have traditionally worked by themselves or in collaboration with other theorists, and they have established broad and deep research track records in the process. Their potential contribution to the ToNC agenda is immense and should not be conditioned on participation in multi-PI, substantially experimental projects. NSF's Computing and Communication Foundations Division (CCF) should support small,

DRAFT

purely theoretical ToNC-research projects, but, ideally, CCF would not be the only source of such support.

Unresolved Issues

Networks play a key role in many sciences, including physics and biology. Statistical physics studies the macroscopic properties of large systems of simple components, which undergo local interactions at the microscopic level. These local interactions define a network on the simple components, and one could say that these physical objects carry out a networked computation. The human brain might also be viewed as a large network carrying out computation: The local interactions of billions of neurons collectively form a brain and are responsible for our experience as humans. Systems biology studies behavior at the subcellular and cellular levels emerging from local interactions of genes and cells. The way these physical and biological systems operate may be analogous to the way global properties of the WWW emerge from changes and interactions at the local level. The structure of complex combinatorial problems and complex behavior and algorithms derives from local constraints and local interactions.

At both ToNC workshops, a considerable amount of time was devoted to the question of whether networks in physics and biology (and other sciences) are a natural part of the ToNC scope, but no firm conclusions were reached. Clearly, computation in physical and biological systems is of interest to Computer Scientists, and good work in this area has been underway in the ToC community for years. However, substantial technical overlap, either in results or in methods, has not been established with the type of research explored in Sections II, III, and IV above. Moreover, most ToNC-workshop participants have never worked on computational questions arising in physical and biological systems and don't foresee doing so in the near- and medium-term future. Concerns were raised about the possibility of defining ToNC so broadly that it becomes meaningless and about crafting funding solicitations so broadly that they result in a deluge of thoroughly incomparable proposals.

The research style evinced in Sections II, III, and IV is very deeply Computer-Scientific. It privileges algorithmic results and algorithmic thinking to an extent not done in the research agendas of other communities that work on networks. The phrase "theory of networked computation" was chosen in part to suggest a primary role for algorithmic thinking, as phrases such as "theory of network design" or "network theory" would not.

As networks continue to grow in importance throughout the technical world, and GENI commands a large fraction of CISE's attention, alternative theoretical-research agendas in networking will be put forth. In particular, approaches that are distinctly less algorithmic will be pursued. The challenge for Computer-Science researchers interested in pursuing the ToNC vision that we have presented will be to embrace the ideas of these other communities when appropriate, to resist embracing them simply in order to get funding, and to compete successfully with alternative "network-theory" visions by effectively articulating the naturalness of an algorithmic approach.

DRAFT

Next Steps

For ToNC research to be as broad and deep as it promises to be, support will have to be obtained from diverse sources. In particular, funding will have to come from all three divisions in the CISE Directorate at NSF, from Federal agencies other than NSF, and from forward-looking IT companies. This is one of the major challenges ahead for ToNC-community leaders. Vigorous advocacy and outreach will be important in meeting this challenge.

Finally, the ToNC community should coordinate and collaborate with the networking-research community, both in advocacy and in research. ToNC researchers can participate in the Global Environment for Network Innovations [GENI] by formulating testable hypotheses about the inherent power and limitations of networks. The architecture-research community is currently wrestling with fundamental questions about the value, costs, and tradeoffs of various networking primitives and abstractions. Very similar questions must be answered in the pursuit of a rigorous Theory of Networked Computation, and GENI will present a unique opportunity to experiment with new networks that have both innovative functionality and rigorous foundations.

DRAFT

VI. References

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

[AHJ+] M. Adler, N. Harvey, K. Jain, R. Kleinberg, and A. Lehman, "On the capacity of Information Networks," in *Proceedings of the 17th Symposium on Discrete Algorithms*, ACM/SIAM, New York/Philadelphia, 2006, pp. 241-250.

[AC] A. Agrawal and M. Charikar, "On the Advantage of Network Coding for Improving Network Throughput," in *Proceedings of the IEEE Information Theory Workshop*, 2004. http://ee-wcl.tamu.edu/itw2004/program/charikar_inv.pdf

[ACLY] R. Ahlswede, N. Cai, S. Li, and R. Yeung. "Network Information Flow," *IEEE Transactions on Information Theory* **46** (2000), pp. 1204-1216.

[ABL] L.von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM* **47** (2004), pp. 55-60.

[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.

[Ande] R. Anderson, "Why Information Security is Hard – An Economic Perspective," invited presentation at the *2001 ACM Symposium on Operating System Principles*. Paper available at http://www.cl.cam.ac.uk/ftp/users/rja14/econ.pdf.

[AS] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys* **36** (2004), pp. 335-371.

[BC] M. Blumenthal and D. Clark, "Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world," *ACM Transactions on Internet Technology* **1** (2001), pp. 70-109.

[Can0] R. Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols," in *Proceedings of the 42nd Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 2001, pp. 136-145.

[Can1] R. Canetti, "Universally Composable Signature, Certification, and Authentication," in *Proceedings of the 17th Computer Security Foundations Workshop*, IEEE Computer Society Press, Los Alamitos, 2004, p. 219-235.

[CKL] R. Canetti, E. Kushilevitz, and Y. Lindell, "On the Limitations of Universally Composable Two-Party Computation Without Set-Up Assumptions," *Journal of Cryptology* **19** (2006), pp. 135-167.

[CLOS] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai, "Universally composable two-party and multi-party secure computation," in *Proceedings of the 34th Symposium on Theory of Computing*, ACM Press, New York, 2002, pp. 494-503.

[Chau] D. Chaum, "Security without Identification: Transaction Systems to Make Big Brother Obsolete," *Communications of the ACM* **28** (1985), pp. 1030-1044.

DRAFT

[CDM+] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, "Towards Privacy in Public Databases," in *Proceedings of the 2nd Theory of Cryptography Conference*, LNCS volume 3378, Springer, Berlin, 2005, pp. 363-385.

[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.

[CGKS] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval," *Journal of the ACM* **45** (1998), pp. 965-982.

[CIPR] Committee on Intellectual Property Rights and the Emergency Information Infrastructure, **The Digital Dilemma: Intellectual Property in the Information Age**, National Academy Press, Washington DC, 2000.

[CDK+] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris, "Practical, Distributed Network Coordinates," *Computer Communication Review* **34** (2004), pp. 113-118.

[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.

[CTA] Cyber-Threat Analytics, http://www.cyber-ta.org/

[DCK+] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A Decentralized Network-Coordinate System," in *Proceedings of Sigcomm '04*, ACM Press, New York, 2004, pp. 15-26.

[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.

[FKP] A. Fabrikant, E. Koutsoupias, and C. Papadimitriou, "Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet," in *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, LNCS volume 2380, Springer, Berlin, 2002, pp. 110-122.

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

[FKSS] J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker, "Hardness Results for Multicast Cost Sharing," *Theoretical Computer Science* **304** (2003), pp. 215-236.

[FPS] J. Feigenbaum, C. Papadimitriou, and S. Shenker, "Sharing the cost of multicast transmissions," *Journal of Computer and System Sciences* **63** (2001), pages 21–41.

[FP+1] J. Feigenbaum, B. Pinkas, R. Ryger, and F. Saint-Jean, "Secure Computation of Surveys," *EU Workshop on Secure Multiparty Protocols*, 2004. http://crypto.stanford.edu/portia/pubs/articles/FPRS1820104872.html

DRAFT

[FP+2] J. Feigenbaum, B. Pinkas, R. Ryger, and F. Saint-Jean, "Some Requirements for Adoption of Privacy-Preserving Data Mining," PORTIA Project White Paper, April 2005. http://crypto.stanford.edu/portia/pubs/articles/FPRS446165839.html

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

[Gasa] W. Gasarch, "A survey of private information retrieval," *The Bulletin of the EATCS* **82** (2004), pp. 72-107.

[GENI] Global Environment for Network Innovations, http://www.geni.net

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

[Gold] O. Goldreich, **The Foundations of Cryptography**, Cambridge University Press, 2004. See volume 2, chapter 7.

[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.

[GJR] T. Griffin, A. Jaggard, and V. Ramachandran, "Design Principles of Policy Languages for Path-Vector Protocols," in *Proceedings of Sigcomm '03*, ACM Press, New York, 2003, pages 61-72.

[GS] V. Guruswami and M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Transactions on Information Theory* **45** (1999), pp. 1757-1767.

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

[KK] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proceedings of the 6th International Conference on Mobile Computing and Networking*, ACM Press, New York, 2000, pp. 243-254.

[Kle0] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM* **46** (1999), pp. 604-632.

[Kle1] J. Kleinberg, "The Convergence of Technological and Social Networks," presented at the NRC Computer Science and Telecommunications Board's 20th Anniversary Celebration, Washington DC, October 2006.

[KSW] J. Kleinberg, A. Slivkins, and T. Wexler, "Triangulation and Embedding Using Small Sets of Beacons," in *Proceedings of the 45th Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 2004, pp. 444-453.

[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.

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

[LPMS] J. Ledlie, P. Pietzuch, M. Mitzenmacher, and M. Seltzer, "Wired Geometric Routing," Submitted for publication. Available in preprint form as Harvard University Computer Science Technical Report TR-19-06, November 2006

DRAFT

[LM+1] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient Erasure Correcting Codes," *IEEE Transactions on Information Theory* **47** (2001), pp. 569-584.

[LM+2] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved Low-Density Parity-Check Codes Using Irregular Graphs," *IEEE Transactions on Information Theory* **47** (2001), pp. 585-598.

[MNPS] D. Mahlki, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay – A Secure, Two-Party Computation System," in *Proceedings of the 13th Symposium on Security*, USENIX, Berkeley, 2004, pp. 287-302.

[Mead] C. Meadows, "The NRL Protocol Analyzer: An Overview," *Journal of Logic Programming* **19-20** (1994), pp.1-19.

[MSVV] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and Generalized On-line Matching," in *Proceedings of the 46th Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, 2005, pp. 264-273.

[MPS] M. Mihail, C. Papadimitriou, and A. Saberi, "On Certain Connectivity Properties of the Internet Topology," in *Proceedings of the 44th Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 2003, pp. 28-35.

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

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

[N] M. Naor, "Verification of a human in the loop or Identification via the Turing Test," Manuscript, 1996. http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.pdf

[NS] V. Narayanan and V. Shmatikov, "Obfuscated Databases and Group Privacy," in *Proceedings of the 12th Conference on Computer and Communications Security,* ACM Press, New York, 2005**,** pp. 102-111.

[NR] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," *Games and Economic Behavior* **35** (2001), pages 166-196.

[Niss] H. Nissenbaum, "Privacy as Contextual Integrity," *Washington Law Review* **17** (2004), pp. 101-139.

[PBMW] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Algorithm: Bringing Order to the Web," Available at http://dbpubs.stanford.edu/pub/1999-66

[Patt] D. Patterson, "20th century vs. 21st century C&C: the SPUR manifesto," *Communications of the ACM* **48** (2005), pp. 15-16.

[PLMS] P. Pietzuch, J. Ledlie, M. Mitzenmacher, and M. Seltzer, "Network-Aware Overlays with Network Coordinates," to appear in *Proceedings of the 1st International Workshop on Dynamic Distributed Systems*, IEEE Computer Society Press, Los Alamitos, 2006. http://www.eecs.harvard.edu/~michaelm/postscripts/iwdds06.pdf

[Rama] V. Ramachandran, *Foundations of Interdomain Routing*, PhD Thesis, Yale University, December 2005. http://www.cs.stevens.edu/~vijayr/papers/dissertation.html

DRAFT

[SRC] J. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Transactions on Computer Systems* **2** (1984), pp. 277-288.

[Sobr] J. Sobrino, "An algebraic theory of dynamic network routing," *IEEE/ACM Transactions on Networking* **13** (2005), pp. 1160-1173.

[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.

[SCE+] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, "HLP: A Next-Generation Interdomain-Routing Protocol," in *Proceedings of Sigcomm '05*, ACM Press, New York, 2005, pp. 13-24.

[Suda] M. Sudan, "Decoding of Reed-Solomon Codes Beyond the Error-Correction Bound," *Journal of Complexity* **13** (1997), pp. 180-193.

[ToNC] Towards a Theory of Networked Computation, http://www.cs.yale.edu/homes/jf/ToNC.html

[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.

[Yao] A. C. Yao. "How to generate and exchange secrets," in *Proceedings of the 27th Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 1986, pp. 162–167.

[Yekh] S. Yekhanin, "New Locally Decodable Codes and Private Information Retrieval Schemes," Electronic Colloquium on Computational Complexity, Report TR06-127, October 2006.

[YMS+] L. Yuan, J. Mai, Z. Su, H. Chen, C.-N. Chuah, and P. Mohapatra, "FIREMAN: A Toolkit for FIREwall Modeling and Analysis," in *Proceedings of the 27th Symposium on Security and Privacy*, IEEE Computer Society Press, Los Alamitos, 2006, pp. 199-213.

DRAFT

Appendix: Workshop Format and Participants

The material in this report 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

DRAFT

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.

DRAFT