Towards a Theory of Networked Computation

1. Introduction

The emergence of large networks like the Web and the Internet is one of the most profound shifts in focus in computer science since its inception. During the last 50 years, computer science has focused primarily on understanding how best to design, build, analyze, and program a computer. The emergence of the Internet and the Web represents a radical shift of focus in our thinking about computational systems. Perhaps the most important distinguishing feature of these networks is that they are simultaneously built, operated, and used by multiple parties with diverse sets of interests and with constantly changing degrees of cooperation and competition. One of the main challenges faced by computer science today is to successfully build and manage large systems created and used by many autonomous organizations and individuals. How can we ensure that such a network functions properly, respects the rights of individual users, and exploits its vast shared resources fully and fairly?

The TCS community can help address the full spectrum of research questions implicit in this grand challenge by developing a coherent Theory of Networked Computation that is analogous to the Theory of (single-machine) Computation that TCS researchers have already developed and continue to pursue fruitfully. The proposed initiative will complement the GENI Initiative and the Cyberinfrastructure program and is close in spirit to Patterson’s SPUR manifesto. TCS 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 coherent Theory of Networked Computation will have great impact on the developing new networked systems, just as formal notions of “efficient solutions” and “hardness” have greatly influenced system development for single machines. The TCS community must leverage the above past achievements and strive to develop a full-fledged Theory of Networked Computation.

To develop such a comprehensive theory, essential features of the Internet would have to be modeled: massive scale, subnetwork autonomy, user self-interest, device heterogeneity, and emergent behavior, among others. Notions of “resources” and “reductions” would help identify the fundamental problems that are not solvable in the context of the Internet, as well as the basic techniques for Internet algorithm design and for expanding the realm of solvable problems. Furthermore, canonically hard problems for networked computation (perhaps a “Cook’s Theorem of Networked Computation”) would not only point to the fundamental obstacles to this methodology, but they could also direct researchers to novel, unanticipated ways of approaching problems. Just as formal proofs of computational hardness can be used in cryptography, hardness in networked computation can possibly lead to system design that makes certain undesirable behaviors (such as cheap mass mailing or denial of service) unattainable within the framework of a given technology.

Note that TCS is in sync with the systems research community in its emphasis on the paradigm shift in computation wrought by the growth of large-scale networks. In an oft-cited CACM President's letter in March 2005, ACM President and UC Berkeley Professor Dave Patterson made a compelling case for the SPUR manifesto:

> In my view, 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. But we now are all painfully aware of the drawbacks of 20th-century C&C. Hence, I believe for our new century we need a new manifesto for C&C: ... Security, Privacy, Usability, and Reliability (SPUR).

This paradigm shift also underlies the GENI initiative. The TCS-research community has a long-standing interest in the fundamental questions inherent in SPUR and GENI, and it played a leading role in identifying, formalizing, and solving some related problems in 20th-century C&C. The community is now expanding and reformulating the notions of security, privacy, usability, and reliability in order to solve the far more complex SPUR-related problems in the networked-computing environment of 21st-century C&C.

In the sections below, we elaborate on some of the research challenges in networked computation. The topics discussed are offered as representatives, not as an exhaustive list. We begin with those put forth by Patterson.

2. Security

Security is unusual among computer-science research areas in that virtually all researchers in the area, including system builders, acknowledge the need for formal models and proofs of security. This is because experimentation alone can never establish that a system is secure. Testing can reveal security flaws, but, once all known flaws have been fixed, there is no guarantee that adversaries will not find more flaws after the system is fielded. Thus, there is great value in proofs that system designs satisfy formal security properties. The TCS community has consistently and energetically responded to this opportunity, developing many fascinating and useful ideas, including zero-knowledge proof systems, digital-signature schemes, and secure hash functions. Furthermore, the community is currently modifying and expanding the theory to better address the scale, complexity, interactivity, and other crucial elements of 21st-century networked computing.

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. For example, following seminal work by Ross Anderson, 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.

3. Privacy

Much of 20th-century computational theory equates "privacy" with confidentiality or secrecy. This approach has proven to be inadequate as more and more sensitive information about people and organizations is created, captured, and stored by the computers and networks that mediate our daily lives. 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 goal is to 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. There are many projects, both ongoing and proposed, in which TCS researchers can contribute to an appropriate technological and conceptual foundation for the handling sensitive information in a modern networked environment.

Helen Nissenbaum's work on "privacy as contextual integrity" posits that a person's privacy is respected in an information environment that respects the "norms of appropriateness" and the "norms of flow" of the cultural context in which the person and the technology are situated. This work has been tremendously successful as a social theory and has attracted the attention of many computer scientists, but the extent to which and the scenarios in which a computational realization of contextual integrity is achievable remain open questions. Two complementary scenarios in which those questions are currently being explored by TCS researchers are the "census" scenario (see recent work on "privacy in public databases") and the "Jet Blue" scenario (see recent work on "group privacy"). The theory of "group privacy" addresses the question of whether a large database can be encoded so that a partially trusted user can retrieve a small set of records that he can describe precisely but cannot retrieve any large set of records and cannot retrieve any set of records that he cannot describe precisely. 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. A complete solution to this problem will have to address the massive scale of modern databases, the interplay of cryptography and coding theory, and the need to limit (often to eliminate completely) interaction between a database owner and a user to whom he has transferred encoded data.

4. Usability

The last five or ten years have seen repeated calls for more realistic system modeling and threat modeling in TCS, and the need to fit human behavior into the models is often mentioned. Both "ease of use" and "hardness of misuse" have been identified as appropriate design criteria for all manner of information systems. Ongoing examples of research efforts that both use TCS methodology and take human behavior into account include the CAPTCHA (Completely

Automated Public Turing test for telling Computers and Humans Apart) technologies that defend websites intended for humans against abuse by bots and recent work on “secure distributed human computation,” in which the goal is to use large-scale networks of **human** agents to solve problems (such as image analysis and natural-language processing) that are difficult for computers (and difficult even for the massive number of computers that work together over today’s Internet). Note that Internet-enabled computation is an essential motivation for both of these projects.

5. Reliability

Much of the above discussion of security carries over *mutatis-mutandis* to reliability. Interestingly, this might be a part of the SPUR agenda in which there is disagreement about Patterson’s formulation of the goal. In his CACM President’s Letter, he says that “21st-century C&C should be as reliable as 20th-century telephony.” In May 2004, Butler Lampson (a towering figure in 20th-century C&C) claimed that public enthusiasm for cell phone service and networked commodity PCs constitutes proof that the old Bell Labs, “5 nines” approach to reliability in 20th-century telephony was overkill; users apparently would have been happy to pay less and get less. TCS researchers can play a crucial role in formulating and solving reliability questions.

A good starting point is the search for appropriate definitions. Reliability *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 “reliable” service or protocol is one that maintains good performance in the presence of system failures and/or adversaries. With such definitions in hand, the natural questions would be whether reliability raises the cost of network services and protocols, if so by how much, and whether users are willing to pay this increased cost. Some results along these lines exist for various notions of “service,” “performance,” “failures,” and “adversaries,” but many interesting questions remain open. Note once again that economics, user self-interest, massive scale, and other essential features of networked computation will drive the development of this theory.

6. Massive-data-set computation

Two of the longest lived, most robust trends in IT are the ever-decreasing cost of data storage and the ever-increasing ubiquity of computers and networks in business, government, recreation, and many other aspects of daily life. Thus, more and more data about people and organizations are created, captured, and stored. The accelerating deployment of sensor networks and surveillance systems will ensure that our ability to create, capture, and store potentially interesting data continues to strain our algorithmic and conceptual ability to understand and use these data. Thus, massive-data-set (MDS) computation will be a central theme of the Networked-Computing research agenda.

The TCS community has already taken up this challenge on multiple fronts. New computational models have been developed, including data streaming, external memory and cache obliviousness, sampling, spot checking, and property testing. The emphasis has been on near-linear, linear, or even sub-linear time and/or space requirements, because the standard TCS notions of polynomial time and space are inadequate for MDS computation. Randomization and approximation are essential in many MDS tasks, and the fact that the TCS community has studied both in depth for many years will stand us in good stead. Powerful and sometimes unexpected connections between MDS computation and other computational paradigms have been discovered, e.g., between communication complexity and streaming and between parallel algorithms and external memory algorithms. Recent MDS algorithmic results have had practical impact in areas such as network measurement, database query optimization and approximate query answering, similarity search, string processing, and geographic information systems.

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. Lower-bound techniques in the new computational models are still few and far between. Security and privacy, strategic and adversarial behavior, and complex data formats (such as images, video, and audio) are aspects of MDS computation that have barely begun to be addressed. The Web-searching problem domain perfectly exemplifies both the great progress that has been made and the tough challenges that lie ahead. On the one hand, 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 will have to be solved if we are to have “personalized search” (which strongly implicates privacy), defenses against “Google bombing” and other adversarial behavior by webpage owners, the ability to search for video or audio clips as well as keywords, and many other Web-based services that truly create value for users.

7. Massive-data-set storage and communication

Along with MDS computation, a full-fledged Theory of Networked Computation will have to consider MDS storage and MDS communication. The amount of data stored on digital media (CDs, DVDs, Zip drives, Computer hard disks) has exploded in the past decade, as has the volume of data communicated via various media (Cell phones, emails, SMS text messages etc.). This increase is expected to continue unabated in the near future. Associated with the increased amount of storage are increased expectations: One hopes to be able to transmit more information, faster, over communication channels and to store more information, cheaply and for longer periods of time, on the storage media. However, all channels introduce noise over time and corrupt the stored/transmitted information. The task of coping with errors in a large-scale, networked C&C environment leads to new challenges and to a resurgence of some classical ones.

The TCS-research community can contribute substantially to progress in MDS storage and communication. On the one hand, TCS researchers have deep algorithmic knowledge, and algorithmic solutions (a relatively recent focus of the more-than-50 year old theory of error correction) will be crucial going forward. Furthermore, the TCS perspective, which links problems from a wide spectrum of areas, is also likely to be crucial to the MDS storage-and-communication agenda. Indeed, for more than a decade, there have been significant contributions by TCS researchers to the theory of reliable communication of information, in the forms of both new algorithms and new models. Examples include the ability to correct more errors, faster error-correction algorithms, rateless codes, checkable codes, list decoding, computationally bounded channels, and the celebrated connection between probabilistically checkable proof systems and nonapproximability of some classical optimization problems.

Still, there are many challenges ahead. Ever-increasing data rates call for still more powerful error-correction techniques and faster algorithms. More significantly, the Internet environment calls for changing the models that researchers traditionally worked with in digital communication. For instance, when dealing with simple media like electrical wires or satellite communications, it was quite reasonable to assume that errors were non-malicious, i.e., that they occurred as a result of some uncontrollable, but simple, probabilistic process. In the new environment, where communication media often are much more complex and involve a chain of links in a dynamic network, oblivious probabilistic error models are not realistic. Furthermore, the communication media and storage media often evolve and change even as the error-correction schemes change. Similar concerns arise in storage, where modern-day media are complex objects such as websites or networks of processors. Such variations in storage and communication technology lead to novel modeling challenges and amplify the need for novel algorithmic solutions.

8. 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: hence the design of incentive-compatible, distributed algorithms. By building explicit payments to computational agents into the protocol, a system designer can incentivize the revelation of relevant private information and the choice of strategies that drive the overall system into a desirable equilibrium state.

The TCS community has focused intently on these issues in recent years. 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. For example, 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)? Can incentive-compatible system designs handle agents with rapidly changing and apparently self-contradictory motivations and utility functions? 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 definitions needed? 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. Whether this approach can be generalized and which system resources can be used as currency in this manner are open questions. All of these open problems fit naturally into our Theory of Networked Computation agenda.

9. Network Computation in Physics and Biology

Networks play a key role in many sciences, especially 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 in a way these physical objects carry out a networked computation. Our brain also can be viewed as a large network carrying out computation. The local interactions of billions of neurons form our brain and are responsible for our experience as humans. Understanding the way the brain works is often viewed at the most important goal in all of science. Systems biology studies behavior at the subcellular and cellular levels emerging from local interactions of genes and cells. The way these biological and physical networks operate is clearly analogous to the way global properties of the WWW emerge from local 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.

Algorithmic paradigms based on the analogy with statistical physics have been spectacularly successful. In systems biology and neuroscience, the behavior of individual elements is somewhat understood, while the behavior of the network is not. Using the random graphs developed in TCS, Les Valiant has made some progress in describing algorithms for a few cortical tasks on such networks, which he views as akin to random graphs. It is likely that one can discover optimum algorithms by a process of reverse engineering --- how would we implement such algorithms on such slow, fault-prone networks? Multidisciplinary research will be crucial in helping us understand how biological networks operate.

10. Verification

As our society's dependence on software systems grows, there is growing awareness of the importance of ensuring that such systems are correct, reliable, and secure, rather than merely loaded with features. The classical approach to ensuring these properties is based on simulation and testing, but this approach detects errors only in the late stages of development, and coverage is only partial. Formal verification research has advocated a more principled approach to system design, using rigorous mathematical specification of what the system is supposed to do, and tool-supported ways of checking whether the design or the implementation meets the specification. Recent years have witnessed remarkable progress in principles and tools for automated verification of hardware and software systems, resulting in adoption of this technology by major industrial players, including Intel (where a large team applies formal verification to processor designs) and Microsoft (which uses software model checking to check conformance of device-driver code). It is undisputed that the technology underlying the tools used by these groups was developed by academic researchers, including members of the TCS community.

Key advances in verification tools are often associated with theoretical insights in logics and automata, sometimes in an unexpected manner and decades later (for example, the notion of Craig interpolants for unsatisfiability cores has recently led to dramatic improvements in refining abstractions of C code for software model checking). Logic offers means for formalizing the requirements of a system. Examples of such logics include the classical Floyd-Hoare logic for sequential programs, temporal and fixpoint logics for reactive programs, and logics tailored for authentication and security properties of cryptographic protocols. Connections between automata over infinite words and trees and temporal logics have proved to be central in formulating and explaining decision procedures. For example, the popular model checker SPIN (which won the ACM Software Systems Award in 2001) employs linear temporal logic as a requirements language, a contribution for which Amir Pnueli received the ACM Turing Award in 1996, and uses an automata-theoretic model-checking algorithm.

The heart of any verification tool is its inference engine. Resolution for propositional logic has a long history, and modern solvers for propositional satisfiability have perfected the resolution engine to a level where they can be used routinely on industrial-scale problem, using hundreds of thousands of variables. Symbolic fixpoint evaluation using binary decision diagrams was developed by formal-methods researchers, was instrumental in fueling industrial interest in model checking, and won the ACM Kannellakis Theory in Practice Award in 1999. Decision procedures for logics with equality, logics with unintereprected functions, and techniques for combining decision procedures are constantly being refined and improved for use in verification tools. Much remains to be done to make these tools or their descendants suitable for Internet-scale problems.

Looking ahead, challenges and opportunities for formal-methods research are abundant. Along with Patterson's SPUR challenge, Hoare's recent grand challenge of verified software is another "call to arms" for revolutionizing the way software is developed. To meet these challenges, the tools need to be effective not just in finding errors, but in proving and certifying correctness. The need for certified software with precise and well understood specifications is particularly critical in emerging applications (e.g., networked, autonomous medical devices) that will distinguish the 21st-century C&C environment. All of this calls for sustained funding for long-term research in principles of formal methods as part of a program in Theory of Networked Computation.

11. Conclusion

We have outlined a sprawling, challenging agenda, the goal of which is fundamental understanding of the capabilities and limitations of our rapidly evolving networked environment In conclusion, we note that Internet pioneer and MIT Senior Research Scientist David Clark (among others) has recently asked how computer scientists would design the Internet architecture today, given all that has been learned since the current design was deployed, if we could build it from scratch? Even without plans for a whole new Internet, research based on this question could lead to important improvements of the current Internet. The theoretical-research agenda that we have proposed could play a central role in answering this question and be an important complement, emphasizing long-term impact, to the GENI Initiative.