A Network Architecture for Large-Scale Science

Sergey Gorinsky **Christoph Jechlitschek** **Nageswara S. V. Rao**

Washington University in St. Louis
One Brookings Drive, Campus Box 1045
St. Louis, MO 63130-4899, USA
{*gorinsky,chrisj*}*@arl.wustl.edu*

Oak Ridge National Laboratory
One Bethel Valley Road, P.O. Box 2008
Oak Ridge, TN 37831-6016, USA
raons@ornl.gov

***Abstract*—Within the framework of the Diversified Internet where architecturally different metanetworks coexist on a shared communication substrate, we present a Large-scale Scientific Metanetwork (LSM) designated for special communication needs of large-scale science. LSM emphasizes performance and security over horizontal scalability and offers two services to users: prompt-delivery service for quick transfers of bulk data and assured-capacity service for applications that require delay or rate guarantees. The prompt-delivery service relies on message-grained scheduling to provide near-minimal average message delay while treating each individual message fairly. In support of the assured-capacity service, LSM maintains differential tree data structures to store and update advance reservations of communication capacities. Explicit accounting for real-world entities involved in communications forms a basis for secure capacity allocation in LSM.**

I. Networking and Large-Scale Science

Needs of large-scale science have driven the field of high-performance computing and yielded designs that are dramatically different from ubiquitous personal computers with respect to both performance and architecture. In this paper, we advocate a view that computer networks for large-scale science should also differ in architecture from the ubiquitous Internet, partly in order to sustain high performance of distributed scientific applications.

Communication requirements of large-scale scientific applications are diverse and range from quick transfers of massive data sets to guarantees of timely delivery for instrument control, collaborative visualization, and remote computational steering [1], [2]. The Internet architecture provides large-scale scientific applications with inadequate support across the whole spectrum of their communication needs. Unrestricted access to the Internet infrastructure precludes guaranteed network services. Even in the context of best-effort delivery under current load conditions, the Internet service is significantly suboptimal [3], [4].

Poor security is another inadequacy that has recently grown into a major concern about the Internet architecture. In particular, efficiency and fairness of Internet capacity usage rests essentially on unwritten rules of ethical behavior. Most traffic shuttles through the Internet over Transmission Control Protocol (TCP) which prescribes how to transmit data as a flow of packets [5]–[7]. However, while a selfish party can relatively straightforwardly change the TCP code at a host, simple changes even in the receiving portion of the code enable the misbehaver to acquire an unfairly high share of the path capacity at the expense of subdued well-behaving cross traffic [8]. Other Internet congestion control protocols are also easily susceptible to manipulation by greedy parties [9]–[11]. Even without any manipulation with underlying transport protocols, misbehaving applications can achieve similar unfair gains by communicating data over multiple concurrent flows [12]. Furthermore, an application has a luxury of disregarding the other consumers of its Internet path capacity by sending data over User Datagram Protocol (UDP) [13] which does not exercise any form of congestion control.

The performance and security deficiencies of the Internet architecture are not an oversight amenable to obvious correction. The deficiencies represent the reverse side of the excellent horizontal scalability exhibited by this generic unassuming architecture. The ability to interconnect arbitrary numbers of devices from various administrative domains with no restrictions on communication media is the primary reason for the stunning ubiquitousness of the Internet. Over the past two decades, a number of alternative network architectures have been advocated. Examples include Integrated Services [14], [15], Differentiated Services [16], and Active Networks [17]. The alternative architectures offered enhanced network services but failed to acquire widespread end-to-end deployment, partly due to being more complex. These experiences indicate a fundamental tension between application-aware network services and scalability.

The imperfections of the incumbent Internet architecture has fed a growing frustration among the large-scale science community that suffered most acutely from the inability of the Internet to translate tremendous capacities of optical communication media into adequate performance of large-scale scientific applications. Early attempts to remedy the status quo focused on automatic tuning of the Internet protocol suite [18]–[21]. However, accumulated empirical evidence revealed that neither modification of the existing Internet transport protocols nor design of new protocols within the Internet framework is likely to yield satisfactory support for the challenging communication needs of large-scale science. Hence, researchers redirected their efforts to designing new types of protocols [22], [23] for dedicated large-scale scientific networks such as UltraScience Net (USN) [24], [25].

1-4244-1580-2/07/$25.00 ©2007 IEEE.

Running independent global or merely wide-area networks is an expensive undertaking which only few entities can afford. Network virtualization [26] offers a more economically viable alternative for resolving the tension between architectural scalability and application-sensitive network services. In the Diversified Internet [27], [28], multiple end-to-end metanetworks coexist on a common substrate of physical communication resources. The diversification abandons the traditional one-size-fits-all paradigm and enables concurrent usage of the communication substrate by architecturally different networks. Such virtualization not only amortizes the costs of operating own end-to-end networks but also (and perhaps more importantly) promotes networking innovation stifled by the current Internet architecture. This paper presents a metanetwork architecture designed for special communication needs of large-scale science.

The rest of the paper is structured as follows. Section II elaborates on the vision of the Diversified Internet. Section III outlines the architecture of the proposed Large-scale Scientific Metanetwork (LSM). Section IV explains how LSM provides meaningful network security by considering real-world entities involved in communications. Section V discusses our preliminary LSM implementation. Finally, Section VI concludes the paper with a summary and discussion of future work.

II. Diversified Internet

The Diversified Internet involves three types of real-world entities: users, metanetwork providers, and infrastructure providers. An infrastructure provider owns and maintains physical communication facilities but does not serve any users directly. Instead, infrastructure providers lease their physical communication resources in the form of metanodes and metalinks to metanetwork providers who supply end-to-end services to users. A physical node can encompass multiple metanodes and hence support multiple metanetworks concurrently. Also, a user can be a customer of multiple metanetworks, e.g., because they offer complementary services.

A key innovation of the Diversified Internet is its separation of delivery services from infrastructure provisioning, which traditional Internet Service Providers (ISPs) bundle together. This separation helps with addressing the conflict between functionality and scalability. While metanetworks are envisioned to be relatively static and few, scalable allocation of physical resources to metanetworks (rather than flows) is a hard but manageable task. For each particular metanetwork, the metanetwork provider solely decides on how to balance scalability versus performance and security of end-to-end services. Global Environment for Network Innovations (GENI) [29] is a planned experimental platform with a large potential for evaluating and realizing the Diversified Internet vision.

III. Large-scale Scientific Metanetwork

Within the framework of the Diversified Internet, we propose a Large-scale Scientific Metanetwork (LSM) designated for special communication needs of large-scale science. Since the Internet diversification does not completely eliminate the tension between scalability and sensitivity to application requirements but merely shifts it to the metanetwork level, each metanetwork has to find own answers to dealing with this conflict. The approach taken in LSM is to favor performance and security over horizontal scalability of the metanetwork architecture. LSM is envisioned to span a wide geographical area, contain only predictable high-capacity metalinks, serve a relatively small number of users at scientific laboratories, and provide optimized secure support for large-scale scientific applications.

LSM classifies its metanodes as edge nodes and core nodes. While core nodes connect only to devices within the network, edge nodes also serve as access points for applications. An application either runs in the secure environment provided by an edge node or is responsible for providing its own secure connection from its remote location to the edge node. LSM annotates its metanodes and metalinks explicitly with such characteristics as capacity, propagation delay, switching type (packet versus circuit), spatial and temporal granularities of the capacity allocation. Explicit accounting for the link properties enables LSM to harvest the full potential of used communication media, e.g., to benefit from statistical multiplexing or end-to-end optical circuits when possible.

In designing the service interface exposed to applications, we strive for effective balance between interface expressiveness and internal complexity needed to realize the interface. The basic abstraction is a logical *channel* from one edge node to another. The architecture permits implementing a channel over multiple physical paths. An application specifies its communication needs by providing LSM with a *request*. We observe that large-scale scientific applications are too diverse to be satisfied with a single class of service. For instance, communication requirements are quite different in real-time visualization of remote computations versus quick transfer of massive data sets. On the other hand, numerous classes of service would weigh down the architecture and undermine its scalability unnecessarily. To achieve the effective balance, LSM offers two types of service to applications:

- *Assured-capacity service* supplies a channel with assured properties such as duration, end-to-end capacity, or propagation delay;
- *Prompt-delivery service* delivers an atomic application-level data unit called a *message* as soon as possible.

With either assured-capacity or prompt-delivery requests, an application is able to specify whether it wants LSM to deliver all sent data reliably.

In consistency with the Diversified Internet vision, LSM comprises a single administrative domain. The single administrative authority not only simplifies metanetwork management but also enables meaningful security, the lack of which hampers the current Internet.

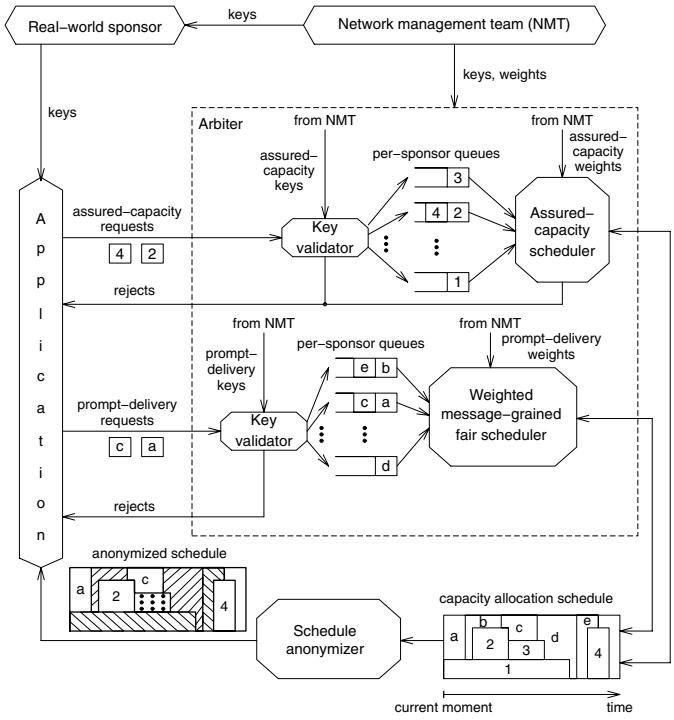


Fig. 1. Network capacity allocation in LSM.

IV. Network Security and Real-world Entities

A prominent feature of LSM is secure allocation of network resources among real-world entities such as individuals and organizations. The traditional approach to network security ignores the active role that real-world entities play in the capacity consumption. Consequently, algorithms for fair end-to-end congestion control or link scheduling allocate resources between packet flows or other purely technical entities. This approach is inherently flawed because flows and processes that generate them are massively replicable by a real-world entity. Even if a network design incorporates mechanisms for validating the identities of flows or processes that generate them, the real-world entity behind an application can subvert the fairness objectives of the design and acquire an unfairly high portion of the network capacity by expressing its communication demands through a multitude of flows or processes.

We remedy this traditional flaw by enriching the metanetwork architecture with an explicit notion of a *real-world sponsor* of an application. The real-world sponsor can be a private individual or a representative of an organization. Applications – e.g., Terascale Supernova [1] and other multi-user applications – can have multiple real-world sponsors. Furthermore, a real-world entity can sponsor multiple applications. LSM allocates its capacity among applications based on their sponsorship by real-world entities. Hence, the ability of a real-world entity to boost the capacity allocation for a favored application reduces to the capability of soliciting extra sponsorship from other authorized real-world entities.

To represent the metanetwork provider who sets policies for LSM usage, authenticates and authorizes sponsors of applications, our architecture includes a notion of a *network management team*. Once again, the network management team can consist of one individual or multiple real-world entities. The architecture enables pursuits of administrative, economic, and other goals in capacity allocation by characterizing real-world sponsors with attributes called *keys* and *weights* and giving control over the attribute values to the network management team. Keys determine who can sponsor an application. Weights describe relative importance of sponsors and direct how the metanetwork resources are shared during contention.

Figure 1 illustrates capacity allocation in LSM. The network management team provides valid keys to the authorized sponsors and arbiter which allocates network resources in response to requests from applications. The sponsors pass their keys to sponsored applications. The arbiter admits for consideration only those requests that contain valid keys. The arbiter translates the admitted requests into a *capacity allocation schedule* according to the sponsor weights supplied by the network management team. To protect privacy of the real-world entities, the arbiter provides the applications with *anonymized* versions of the capacity allocation schedule.

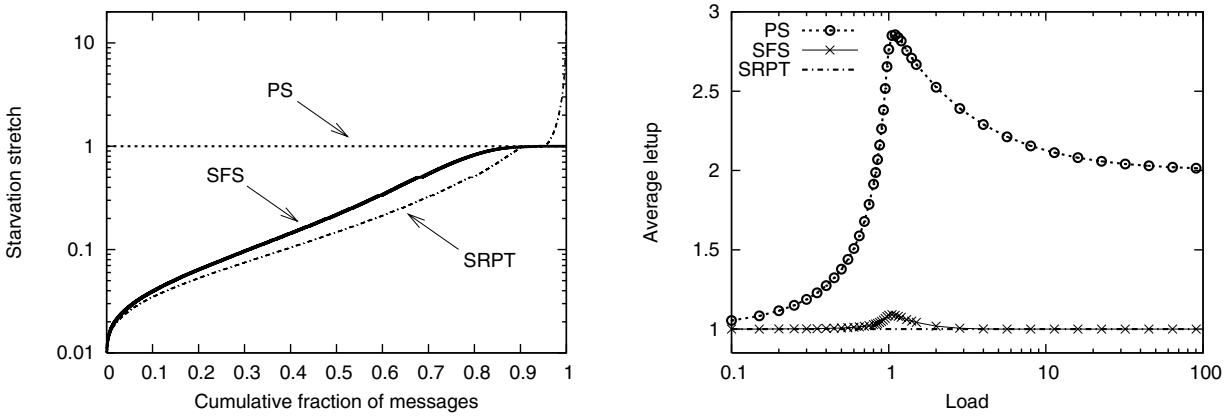
Sponsor keys and weights are control knobs of the arbiter. This external guidance by the network management team is optional. The arbiter can operate autonomously with default settings of keys and weights, such as equal weights for all real-world sponsors from a list authorized at the time of configuration.

The existence of two service classes brings out the question what constitutes a fair resolution of conflicting requests, both within and between the classes. To address the challenge of inter-class fairness, we identify the following preferences of application requests:

- *Assured-capacity requests* favor delayed acquisition of assured network resources over immediate communication at an unpredictable rate;
- *Prompt-delivery requests* prefer the fastest delivery of their messages, even if conducted at an unpredictable rate.

Hence, our arbiter uses time for resolving inter-class resource contentions: wait of assured-capacity requests is balanced against completion time of prompt-delivery requests. Since LSM strives to honor all granted assured-capacity requests, potential unfairness to future requests is tackled by keeping a fraction of network resources unavailable for reservation. The fraction is larger for times further away from the current moment. As the latter advances, the arbiter releases preserved resources. Instead of an outright reject due to unavailability of requested resources, the assured-capacity request receives an option of being automatically reconsidered as preserved resources are released.

The LSM architecture also incorporates mechanisms for network monitoring and protection against attacks. The monitoring mechanism collects resource usage statistics and warns the network management team and users about abnormal situations. Protection mechanisms include access control and



(a) fairness to individual messages in 100 experiments at 95% load

(b) average efficiency

Fig. 2. Message-grained scheduling for the prompt-delivery service: SFS versus SRPT and PS.

rate limiting that prevent a misbehaving party from using LSM in violation of the capacity allocation schedule.

V. LSM Implementation

To implement the secure allocation of LSM resources between real-world entities, we borrow a technique of tokens from USN [24]. An authorized sponsor receives unforgeable tokens according to the sponsor's weight and is free to distribute the tokens among sponsored applications in an arbitrary manner. An application collects tokens from its sponsors and distributes the tokens among its requests supplied to the arbiter. Based on tokens provided with a request, the arbiter decides whether and when the request is granted. An acceptance of a request indicates start time and bitrate for the allocated channel. To support efficient utilization of network resources, applications are allowed to transmit opportunistically in excess of the allocated channel rate. However, if excessive transmission causes contention at a link, traffic beyond the allocated rate is served from a low-priority queue. To secure the network against unauthorized traffic, edge nodes receive random seeds from the arbiter and apply one-way hash chains [30], [31] to verify that forwarded traffic corresponds to granted requests.

The prompt-delivery service is primarily geared toward quick transfer of long messages. Fur such transfers, the capacity of a bottleneck link on the path from the source to the destination is the main factor determining minimal achievable delays. Hence, we seek to build the service around a fair efficient algorithm for allocating the bottleneck link capacity. While Shortest Remaining Processing Time (SRPT) schedules messages preemptively in the order of their remaining transmission delays and is optimally efficient [32], the minimal average delay comes at the expense of potential unfairness: SRPT starves large messages by delaying them without bound in some settings with heavy load [33]. Processor Sharing (PS) achieves fairness by instantaneously allocating equal shares of the bottleneck capacity to all pending messages [34]. However, average delay of messages under PS is significantly higher. Recent research unveiled a class of fair message-grained algorithms that are significantly more efficient than PS and sustain average delay close to the minimum attained under the unfair SRPT [3], [4], [35]. A specific scheme is our Shortest Fair Sojourn (SFS) that schedules a shortest message unless this precludes other pending messages from finishing before their PS completion times; in the latter case, SFS schedules a message with the closest PS finish time [4].

Figure 2 reports results for simulations with 3,000 messages over a 10-Tbps path. Message sizes are uniformly distributed between 100 GB and 100 TB. The messages arrive according to a Poisson process with such an average rate that yields desired load. Figure 2a shows a cumulative distribution of starvation stretch, i.e., the ratio of message delay to message delay under PS. Under SFS, all messages finish no later than under PS, and most of them complete much earlier. Figure 2b plots average letup, the ratio of average delay to average delay under SRPT. In contrast to PS, SFS significantly reduces average delay and consistently provides SRPT-like efficiency.

The prompt-delivery service consumes capacity remained after granted assured-capacity requests. To support advance reservations for assured-capacity requests, the arbiter maintains tree data structures that represent availability of link capacities [36]–[39]. Each vertex of the differential trees contains a three-tuple describing a time interval with the start of the time interval, maximum extra capacity available across the interval, and maximum extra capacity within a subinterval. For example, the available capacity depicted in Figure 3 might be represented with an eight-node three-level tree where the root summarizes the capacity availability before and after time 14 with tuples (0, 4, 4) and (14, 2, 8) respectively. The differential trees enable the arbiter to check for capacity availability and update the reservation schedule with logarithmic time complexity.

VI. Conclusion

Within the framework of the Diversified Internet where architecturally different metanetworks coexist on a shared communication substrate, we proposed LSM, a metanetwork

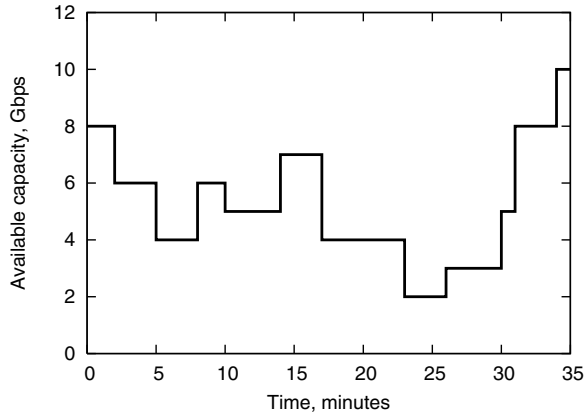


Fig. 3. Sample profile of capacity availability for assured-capacity requests.

for large-scale science. LSM emphasizes performance and security over horizontal scalability. Explicit accounting for real-world entities involved in communications forms a basis for secure capacity allocation. LSM offers two services to users: prompt-delivery service for quick transfers of bulk data and assured-capacity service for applications that require delay or rate guarantees.

Our LSM implementation clearly needs further work. We plan to extend our algorithms to handle general network topologies with multiple bottleneck links. For the prompt-delivery service, a likely consequence is a relaxed fairness criterion. With respect to the assured-capacity service, LSM will offer several resolutions of the advance reservation process, ranging from quick approximate check to thorough search for availability with flexible start times for reserved channels.

REFERENCES

[1] "Terascale Supernova Initiative," http://www.phy.ornl.gov/tsi/.

[2] "DOE Science Networking: Roadmap to 2008," http://www.es.net/hypertext/welcome/pr/Roadmap/, June 2003.

[3] S. Gorinsky and N. S. V. Rao, "Dedicated Channels as an Optimal Network Support for Effective Transfer of Massive Data," in *Proceedings IEEE High-Speed Networking (HSN 2006)*, April 2006.

[4] S. Gorinsky and C. Jechlitschek, "Fair Efficiency, or Low Average Delay without Starvation," in *Proceedings International Conference on Computer Communications and Networks (ICCCN 2007)*, August 2007.

[5] "Transmission Control Protocol," USC Information Sciences Institute, RFC 793, September 1981.

[6] V. Jacobson, "Congestion Avoidance and Control," in *Proceedings ACM SIGCOMM 1988*, August 1988.

[7] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, April 1999.

[8] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson, "TCP Congestion Control with a Misbehaving Receiver," *ACM Computer Communication Review*, vol. 29, no. 5, pp. 71–78, October 1999.

[9] S. Gorinsky, S. Jain, and H. Vin, "Multicast Congestion Control with Distrusted Receivers," in *Proceedings Networked Group Communication (NGC 2002)*, October 2002.

[10] S. Gorinsky, S. Jain, H. Vin, and Y. Zhang, "Design of Multicast Protocols Robust against Inflated Subscription," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 249–262, April 2006.

[11] M. Georg and S. Gorinsky, "Protecting TFRC from a Selfish Receiver," in *Proceedings Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005)*, October 2005.

[12] T. Hacker, B. Noble, and B. Athey, "The Effects of Systemic Packet Loss on Aggregate TCP Flows," in *Proceedings IEEE/ACM Supercomputing*, November 2002.

[13] J. Postel, "User Datagram Protocol," RFC 768, October 1980.

[14] D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," in *Proceedings ACM SIGCOMM 1992*, August 1992.

[15] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994.

[16] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.

[17] D. Wetherall, U. Legedza, and J. Guttag, "Introducing New Internet Services: Why and How," *IEEE Network*, vol. 12, no. 3, pp. 12–19, May-June 1998.

[18] W. Feng, M. Gardner, M. Fisk, and E. Weigle, "Automatic Flow-Control Adaptation for Enhancing Network Performance in Computational Grids," *Journal of Grid Computing*, vol. 1, no. 1, pp. 63–74, March 2003.

[19] M. Gardner, S. Thulasidasan, and W. Feng, "User-Space Auto-Tuning for TCP Flow Control in Computational Grids," *Computer Communications*, vol. 27, no. 14, pp. 1364–1374, September 2004.

[20] "ORNL Net100," http://www.csm.ornl.gov/~dunigan/net100/.

[21] N. S. V. Rao and W. Feng, "Performance Trade-Offs of TCP Adaptation Methods," in *Proceedings IEEE International Conference on Networks (ICN 2002)*, August 2002.

[22] N. S. V. Rao, Q. Wu, and S. S. Iyengar, "On Throughput Stabilization of Network Transport," *IEEE Communications Letters*, vol. 8, no. 1, pp. 66–68, January 2004.

[23] Q. Wu and N. S. V. Rao, "A Class of Reliable UDP-Based Transport Protocols Based on Stochastic Approximation," in *Proceedings IEEE INFOCOM 2005*, March 2005.

[24] N. S. V. Rao, W. R. Wing, S. M. Carter, and Q. Wu, "UltraScience Net: Network Testbed for Large-Scale Science Applications," *IEEE Communications*, vol. 43, no. 11, pp. S12–S17, November 2005.

[25] "DOE UltraScienceNet: Experimental Ultra-Scale Network Testbed for Large-Scale Science," http://www.csm.ornl.gov/ultranet.

[26] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet Impasse through Virtualization," *IEEE Computer*, vol. 38, no. 4, pp. 34–41, April 2005.

[27] J. S. Turner and D. E. Taylor, "Diversifying the Internet," in *Proceedings IEEE Globecom 2005*, November 2005.

[28] N. Feamster, L. Gao, and J. Rexford, "How to Lease the Internet in Your Spare Time," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 61–64, January 2007.

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

[30] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, November 1981.

[31] A. Perrig, R. Canetti, D. Song, and D. Tygar, "Efficient and Secure Source Authentication for Multicast," in *Proceedings Network and Distributed System Security Symposium (NDSS 2001)*, February 2001.

[32] L. E. Schrage, "A Proof of the Optimality of the Shortest Remaining Processing Time Discipline," *Operations Research*, vol. 16, no. 3, pp. 687–690, May-June 1968.

[33] M. A. Bender, S. Chakrabarti, and S. Muthukrishnan, "Flow and Stretch Metrics for Scheduling Continuous Job Streams," in *Proceedings ACM-SIAM SODA 1998*, January 1998.

[34] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987.

[35] E. J. Friedman and S. G. Henderson, "Fairness and Efficiency in Web Server Protocols," in *Proceedings ACM SIGMETRICS 2003*, June 2003.

[36] J. S. Turner, "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3–16, March 1999.

[37] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient Burst Scheduling Algorithms in Optical Burst Switched Networks Using Geometric Techniques," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 9, pp. 1796–1811, November 2004.

[38] C. Castillo, G. Rouskas, and K. Harfoush, "On the Design of Online Scheduling Algorithms for Advance Reservations and QoS in Grids," in *Proceedings IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)*, March 2007.

[39] ——, "Efficient QoS Resource Management for Heterogeneous Grids," submitted, April 2007.