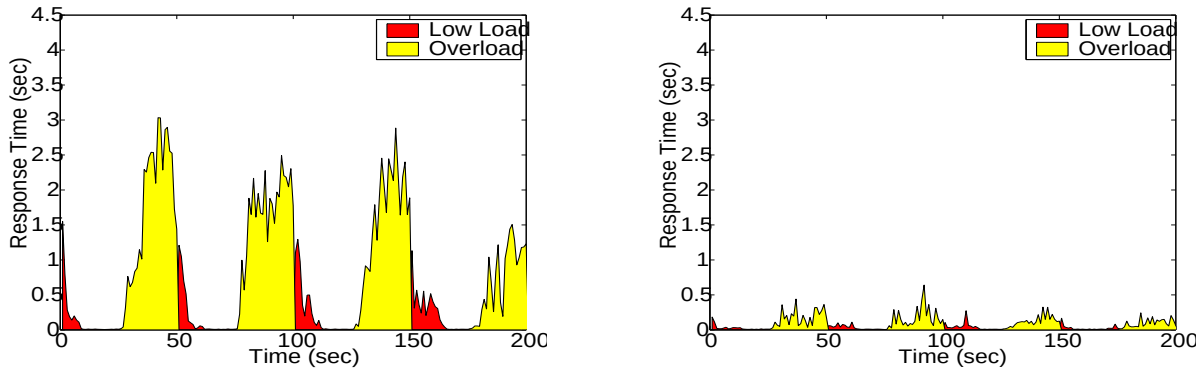
2 Computer Systems with Fluctuating Load

The problem While queueing theoretic analysis is grounded in the notion of stationary load and steady-state behavior, the load on real computer systems is far from stationary. Since web traffic is bursty and hard to predict, even well-provisioned servers can experience *transient (temporary) overload*, leading to long response times and even longer recovery times, even when the *average load* at the server is not very high. Unfortunately, extremely little is understood about systems with non-stationary load. It is not understood how *input parameters* like the arrival rate, service rate, rate of fluctuation in the load, and degree of fluctuation in the load, affect response time. For example, it is not even clear whether increasing the rate of fluctuation in load always causes mean response time to increase.

Analytical difficulty Systems with time-varying load are very difficult to analyze. Techniques used are typically *numerical*, including either Matrix-Analytic techniques, or generating function techniques which rely on numerically solving cubic equations. See for example [1, 10, 17, 11, 16, 21]. Unfortunately, these *don't lead to closed-form approximations* of the system, and hence it is hard to understand the effect of input parameters on performance.

Surprising results – analysis In [3] we present the first *closed-form approximation of mean response time* for a First-Come-First-Served (FCFS) queue with stochastically-fluctuating load (including possibly transient overload), and in [4] we employ a completely different technique to analyze *mean and variability of response time* as a function of job size, for the case of a Processor-Sharing (PS) queue with time-varying load. The impact of these closed-form approximations is that we can immediately understand how input parameters like arrival rate, service rate, and load fluctuations, affect performance. We find many surprises. For example we prove that, counter to prior conjectures [13, 12, 9], increasing the rate of fluctuation in load *does not* always lead to increased response time [3], and we define a simple criterion called *slack*, which tells us whether response time goes up or down. We can also, for the first time, understand the experience of a job arriving into a high-load period, as compared to an arbitrary job.



(a) Unmodified (PS) server (b) SRPT server

Figure 1: *Mean response times at Web server as load alternates between 0.2 and 1.2: (a) under* PS*, (b) under* SRPT.

Surprising results – systems Most systems research does not study the behavior of servers during overload, but rather prefers to look for solutions to obviate overload by moving requests to other servers, as in content distribution networks [8, 7, 6], or by dropping requests, as in admission

control [2, 5, 19, 18, 20]. By contrast, we take the perspective that transient overload is sometimes unavoidable, and we seek solutions that don't require dropping/moving requests. In our award paper [14], we study exactly what happens to a web server during overload, evaluated under a full range of environmental conditions including: the effect of WAN delay, loss, user aborts, persistent connections, SYN cookies, the RTO TCP timer, the packet length, and the SYN and ACK queue lengths.

Figure 1(a) shows the mean response time under an unmodified Apache web server running on Linux, employing traditional PS scheduling, where load fluctuates between 0.2 (low load) and 1.2 (overload). In Figure 1(b), we show how our instrumentation of SRPT scheduling at the Linux kernel reduces mean response times by an order of magnitude, *without dropping any requests*. We see that the SRPT server is much more efficient than the unmodified server at clearing requests out of the system during overload, resulting in a smaller backlog when the overload period ends. This is especially true under heavy-tailed web workloads. Finally, unfairness to large requests is not a problem here either, in that requests for large files complete at similar times under PS and SRPT, [14, 15].

Funding This research was supported by (i) NSF ITR-0313148 "Improving the Performance of Web Servers under Overload" (2003-2007); (ii) Pittsburgh Digital Greenhouse grant 2001-2002; (iii) IBM graduate fellowship/support.

References

[1] E. Arjas. On the use of a fundamental identity in the theory of semi-Markov queues. *Adv. Appl. Prob.*, 4:271–284, 1972.

[2] L. Cherkasova and P. Phaal. Session based admission control: A mechanism for improving the performance of an overloaded web server. Available at http://www.hpl.hp.com/techreports/98/HPL-98-119.html, 1998.

[3] V. Gupta, M. Harchol-Balter, A. Scheller-Wolf, and U. Yechiali. Fundamental characteristics of queues with fluctuating load. In *ACM Sigmetrics 2006 Conference on Measurement and Modeling of Computer Systems*, 2006.

[4] R. C. Hampshire, M. Harchol-Balter, and W. A. Massey. Fluid and diffusion limits for transient sojourn times of processor sharing queues with time varying rates. *QUESTA*, 53(1/2), June 2006.

[5] R. Iyer, V. Tewari, and K. Kant. Overload control mechanisms for web servers. In *Workshop on Performance and QoS of Next Generation Networks*, November 2000.

[6] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek. The measured performance of content distribution networks. In *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, 2000.

[7] B. Krishnamurthy, C. Wills, and Y. Zhang. The use and performance of content distribution networks. In *Internet Measurement Workshop. Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop.*, 2001.

[8] M.Day, B.Cain, G.Tomlinson, and P.Rzewski. A model for content internetworking (cdi). Internet Draft (draft-ietf-cdi-model-02.txt), May 2002.

[9] N. Miyoshi and T. Rolski. Ross-type conjectures on monotonicity of queues. *Australian & New Zealand J. of Stat.*

[10] M. Neuts. The M/M/1 queue with randomly varying arrival and service rates. *OPSEARCH*, 15(4):139–168, 1978.

[11] V. Ramaswami. The N/G/1 queue and its detailed analysis. *Adv. Appl. Prob.*, 12:222–261, 1980.

[12] T. Rolski. Queues with non-stationary input stream: Ross's conjecture. *Adv. Appl. Prob.*, 13:603–618, 1981.

[13] S. Ross. Average delay in queues with non-stationary Poisson arrivals. *J. Appl. Prob.*, 15:602–609, 1978.

[14] B. Schroeder and M. Harchol-Balter. Web servers under overload: How scheduling can help. In *18th International Teletraffic Congress*, 2003. Best Paper Award.

[15] B. Schroeder and M. Harchol-Balter. Web servers under overload: How scheduling can help. *ACM Transactions on Internet Technologies*, 6(1), February 2006.

[16] T. Takine, Y. Matsumoto, T. Suda, and T. Hasegawa. Mean waiting times in nonpreemptive priority queues with Markovian arrival and i.i.d. service processes. *Perf. Eval.*, 20:131–149, 1994.

[17] T. Takine and B. Sengupta. A single server queue with service interruptions. *QUESTA*, 26:285–300, 1997.

[18] T. Voigt and P. Gunnigberg. Kernel-based control of persistent web server connections. *ACM SIGMETRICS Performance Evaluation Review*, 29(2):20–25, 2001.

[19] T. Voigt, R. Tewari, D. Freimuth, and A. Mehra. Kernel mechanisms for service differentiation in overloaded web servers. In *Proceedings of the USENIX Annual Technical Conference*, Boston, MA, June 2001.

[20] M. Welsh and D. Culler. Adaptive overload control for busy internet servers. In *Proceedings of the 2003 USENIX Symposium on Internet Technologies and Systems*, 2003.

[21] U. Yechiali and P. Naor. Queueing problems with heterogeneous arrivals and service. *OR*, 19(3):722–734, 1971.