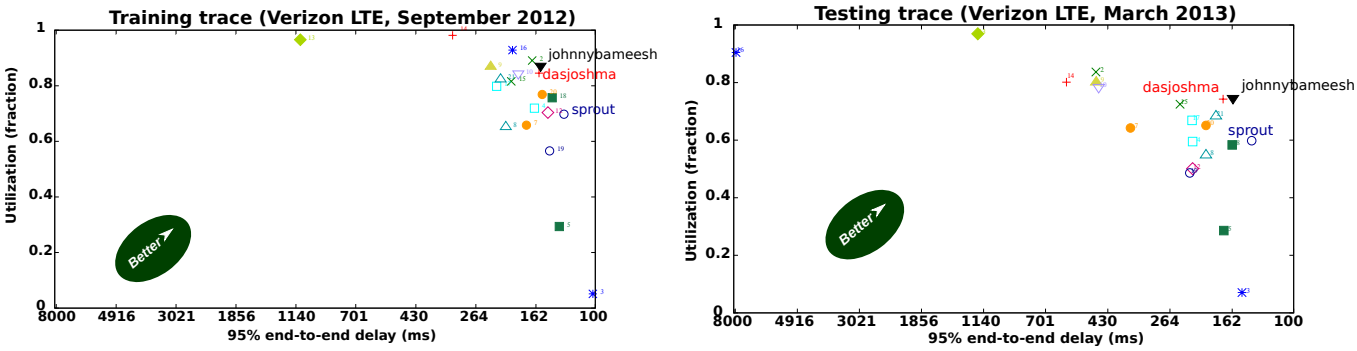

(a) Performance on training trace

(b) Performance on testing trace

Figure 8: Student protocols cluster on the training trace and disperse on the testing trace

1. **Crowdsourced Protocol Design.** Tapping into a larger pool of human resources allows us to design better protocols, assuming an objective evaluation metric is known a priori. Crowdsourcing has largely been used for repetitive and tedious tasks by previous systems such as Mechanical Turk. Design contests allow us to use the wisdom of the crowd for considerably more involved problems.

2. **Reproducible Research.** Turning a research problem into a well-specified contest forces the researcher to clearly articulate the testing conditions and ensure that her protocol works reproducibly under those conditions. This, in turn, makes the protocol accessible to a wider audience of other researchers.

3. **Explicitness in problem specification.** "Gamifying" a research problem entails setting up a well-defined problem where the end objective is clearly specified. Several protocols today (for instance, TCP Cubic [8]) try to do something reasonable under all conditions without specifying their end goal. Specifying an objective explicitly has several benefits:

   (a) It affords a more directed search for the solution.

   (b) It forces the designer to think hard about whether the objective is one that an application truly cares about. The distinction between per-packet delay and end-to-end delay, discussed earlier, is a case in point.

4. **Achievable Region Estimation.** The achievable region (in our case, all achievable (throughput, delay) tuples) is mathematically intractable in several cases. Turning such problems into contests allows us to trace out a crowdsourced achievability region, assuming a large number of submissions.

Such contests have benefited several other fields and we think they confer similar benefits on Computer Networking as a whole. They also provide a means to engage students directly in the activity of research by simplifying a problem down to its essence such that it can be approached by a student with little domain expertise. Feedback from students taking the class was overwhelmingly positive, with several students singling out the contest for praise.

Code for all components of our evaluation infrastructure is available at http://web.mit.edu/anirudh/www/contest.html. In describing our experiences running a congestion-control protocol design contest, we hope to prod others into running similar contests centered around other classical, well-defined problems within networking such as routing, traffic engineering, and scheduling.

6. ACKNOWLEDGMENTS

We thank the 6.829 students for making this contest possible. Jonathan Perry, as the teaching assistant for 6.829, gave us valuable feedback on the design of the contest. We thank the anonymous CCR reviewers and Amy Ousterhout for feedback that significantly improved the quality of the paper.

7. REFERENCES

[1] DARPA Spectrum Challenge. http://www.darpa.mil/spectrumchallenge/.

[2] Kaggle:Go from Big Data to Big Analytics. https://www.kaggle.com/.

[3] Reproducing Network Research — network systems experiments made accessible, runnable, and reproducible. http://reproducingnetworkresearch.wordpress.com/.

[4] Text Analysis Conference (TAC) KBP 2013 Tracks. http://www.nist.gov/tac/2013/KBP/.

[5] D. Bansal and H. Balakrishnan. Binomial Congestion Control Algorithms. In *INFOCOM*, 2001.

[6] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *SIGCOMM*, 1994.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *SIGCOMM*, 2000.

[8] S. Ha, I. Rhee, and L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating System Review*, 42(5):64–74, July 2008.

[9] V. Jacobson. Berkeley TCP Evolution from 4.3-tahoe to 4.3-reno . In *Proceedings of the Eighteenth Internet Engineering Task Force*, 1990.

[10] B. Lantz, B. Heller, and N. McKeown. A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6, New York, NY, USA, 2010. ACM.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A Compound TCP Approach for High-speed and Long Distance Networks. In *INFOCOM*, 2006.

[12] K. Winstein, A. Sivaraman, and H. Balakrishnan. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *USENIX Symposium on Networked Systems Design and Implementation*, Lombard, IL, April 2013.