Internet Congestion: A Laboratory Experiment

Daniel Friedman
University of California
Economics Department
Santa Cruz CA
831.429-8861
dan@ucsc.edu

Bernardo Huberman
Hewlett-Packard
Laboratories
1501 Page Mill Road
Palo Alto CA
650.857-1501
huberman@hpl.hp.com

ABSTRACT
Human players and automated players (bots) interact in real time in a congested network. A player's revenue is proportional to the number of successful "downloads" and his cost is proportional to his total waiting time. Congestion arises because waiting time is an increasing random function of the number of uncompleted download attempts by all players. Surprisingly, some human players earn considerably higher profits than bots. Bots are better able to exploit periods of excess capacity, but they create endogenous trends in congestion that human players are better able to exploit. Nash equilibrium does a good job of predicting the impact of network capacity and noise amplitude. Overall efficiency is quite low, however, and players overdissipate potential rents, i.e., earn lower profits than in Nash equilibrium..

Categories and Subject Descriptors
C.4 PERFORMANCE OF SYSTEMS; J.4 SOCIAL AND BEHAVIORAL SCIENCES

General Terms
Economics, Reliability, Experimentation, Human Factors,.

Keywords
Asynchronous, congestion, automated agents, human subjects.

1. INTRODUCTION
The Internet suffers from bursts of congestion that disrupt cyberspace markets. Some episodes, such as gridlock at the Victoria's Secret site after a Superbowl advertisement, are easy to understand, but other episodes seem to come out of the blue. Of course, congestion is also important in many other contexts. For example, congestion sometimes greatly degrades the value of freeways, and in extreme cases (such as burning nightclubs) congestion can be fatal. Yet the dynamics of congestion are still poorly understood, especially when (as on the Internet) humans interact with automated agents in real time.

In this paper we study congestion dynamics in the laboratory using a multiplayer interactive video game called StarCatcher. Choices are real-time (i.e., asynchronous): at every instant during a two minute period, each player can start to download or abort an uncompleted download. Human players can freely switch back and forth between manual play and a fully automated strategy. Other players, called bots, are always automated. Players earn revenue each time they complete the download, but they also accumulate costs proportional to waiting time.

Congestion arises because waiting time increases stochastically in the number of pending downloads. The waiting time algorithm is borrowed from [8], who simulate bot-only interactions. This study and earlier studies show that congestion bursts arise from the interaction of many bots, each of whom reacts to observed congestion with some short but inevitable lag. The intuition is that bot reactions are highly correlated, leading to non-linear bursts of congestion.

Several other strands of empirical literature relate to our work. [9], [10] and other studies find that human subjects are remarkably good at coordinating entry into periodic (synchronous) laboratory markets subject to congestion, but do not examine whether the results extend to more challenging asynchronous environments with bots. [3] mentions research by Dave Cliff at HP Labs Bristol intended to develop bots that can make profits in major financial markets that allow asynchronous trading. The article also mentions the widespread belief that automated trading strategies provoked the October 1987 stock market crash.

[5] adapts periodic laboratory software to create a near-asynchronous environment where some subjects can update choices every second; other subjects are allowed to update every 2 seconds or every 30 seconds. The subjects play quantity choice games (e.g., Cournot oligopoly) in a very low information environment: they know nothing about the structure of the payoff function or the existence of other players. Play tends to converge to the Stackelberg equilibrium (with the slow updaters as leaders) rather than to the Nash equilibrium. In our setting, by contrast, there is no clear distinction between Stackelberg and Nash, subjects have asynchronous binary choices at endogenously determined times, and they compete with bots.

After describing the laboratory set up in the next section, we sketch theoretical predictions derived mainly from Nash equilibrium. Section 4 presents the results of our experiment. Surprisingly, some human players earn considerably higher profits than bots. Bots are better able to exploit periods of excess capacity, but they create endogenous trends in congestion that

SIGCOMM'04 WORKSHOPS, SEPT. 3, 2004, PORTLAND, OR, USA.

human players are better able to exploit. Nash equilibrium comparative statics do a good job of predicting the impact of network capacity and noise amplitude. Overall efficiency is quite low: relative to Nash equilibrium, players "overdissipate" potential rents.

Section 5 offers some perspectives and suggestions for follow up work. Appendix A collects the details of algorithms and mathematical derivations. Instructions to subjects and the user interface are available at http://leeps.ucsc.edu/leeps/papers.

2. THE EXPERIMENT

The experiment was conducted at UCSC's LEEPS lab. Each session lasts about 90 minutes and typically employs 4-6 human subjects, most of them UCSC undergraduates. Students sign up on line after hearing announcements in large classes, and are notified by email about the session time and place, using software developed by UCLA's CASSEL lab. Subjects read the instructions, view a projection of the user interface, participate in practice periods, and get public answers to their questions. Then they play 16 or more periods of the StarCatcher game. At the end of the session, subjects receive cash payment, typically $15 to $25. The payment is the total points earned in all periods times a posted payrate, plus a $5.00 show-up allowance.

Each StarCatcher period lasts 240 seconds. At each instant, any idle player can initiate a service request by clicking the Download button, as in Figure 1. An algorithm (defined in Appendix A) determines the latency (i.e., the service delay), based on the current number of active download requests, the assigned network capacity that period, and random background noise. Unless the download is stopped earlier, the player's screen flashes a gold star and awards her 10 points when the latency ends. However, each second of delay costs the player 2 points, so she loses money on download requests with latencies greater than 5 seconds. The player can't begin a second download while an earlier request is still being processed but she can click the Stop button; to prevent excessive losses the computer automatically stops a request after 10 seconds. The player can also click the Reload button, which is equivalent to Stop together with an immediate new download request, and can toggle between manual mode (as just described) and automatic mode (described below).

The player's timing decision is aided by a real-time display showing the results of all service requests terminating in the previous 10 seconds. The player sees the mean latency as well as a latency histogram that includes Stop orders.

The experiments include automated players (called robots or bots) as well as humans. The basic algorithm for such players is: initiate a download whenever the mean latency (shown on all players' screens) is less than 5 seconds minus a tolerance, i.e., whenever it seems sufficiently profitable. The tolerance averages 0.5 seconds, corresponding to an intended minimum profit margin of 1 point per download. Appendix A includes the full algorithm. Human players in most sessions have the option of "going on autopilot" using this algorithm.

The network capacity C and the persistence and amplitude of the background noise is controlled at different levels in different periods, and the number of human players and bots also varies as shown in Table 1.

Table 1. Experimental Design

Volatility low: •=.0015, •=.0002; volat. hi: •=.0025, • =.00002. The notation # bot and # hu refer to the maximum number of automated agents (bots) and human subjects in any period, and # per is the number of periods in the session. Exp indicates whether the human subjects were experienced, i.e., had participated in a previous session.

Date of session	# per	# of periods played										# bot	# hu	Exp
		Total	By volat		By capacity C									
			low	hi	2	3	4	5	6	7	9			
8/21/02	27	159	87	72	20	101	38	0	0	0	0	4	4	no
8/22/02	32	189	94	95	19	120	50	0	0	0	0	4	4	no
8/20/02	32	192	97	95	21	117	54	0	0	0	0	4	4	yes
9/11/02	32	243	130	113	0	56	90	0	97	0	0	0	6	yes
9/12/02	32	199	101	98	20	126	53	0	0	0	0	5	3	yes
9/5/02	32	193	99	94	20	121	52	0	0	0	0	4	4	no
1/24/03	16	127	54	73	0	37	40	0	50	0	0	4	6	no
1/31/03	24	155	77	78	20	105	30	0	0	0	0	4	4	no
2/5/03	27	216	120	96	0	54	72	0	90	0	0	4	6	yes
2/4/03	16	104	52	52	10	64	30	0	0	0	0	4	4	no
2/12/03	18	143	71	72	0	36	47	0	60	0	0	4	6	no
2/14/03	27	214	119	95	0	54	72	0	88	0	0	4	6	no
2/19/03	31	194	100	94	19	123	52	0	0	0	0	4	4	yes
5/23/03	27	164	89	75	0	94	64	6	0	0	0	5	6	no
10/2/03	31	112	63	49	76	22	6	8	0	0	0	4	4	no
10/3/03	27	164	86	78	17	45	50	8	0	20	24	6	6	no

3. THEORETICAL PREDICTIONS

A player's objective each period is to maximize profit $P = rN - cL$, where r is the reward per successful download, N is the number of successful downloads, c is the delay cost per second, and L is the total latency time summed over all download attempts in that period. The relevant constraints include the total time T in the period, the network capacity C, and the time scale S.

An important benchmark is social value V^*, the maximized sum of players' profits. Appendix A shows that, ignoring random noise, $V^*=0.25S^{-1}Tr(1+C - cS/r)^2$. Typical parameter values in the experiment are T=120 seconds, C=6 users, S=8 user-sec, c=2 points/sec and r=10 points. The corresponding social optimum values are U^* = 2.70 active users, l^* =1.86 seconds average latency, p^* = 6.28 points per download, N^*=174.2 downloads, and V^*= 1094 points per period.

Of course, a typical player tries to increase his own profit, not social value. A selfish and myopic player will attempt to download whenever the incremental apparent profit p is sufficiently positive, i.e., whenever the reward r=10 points sufficiently exceeds the cost λc at the currently displayed average latency l. Thus such a player will choose a latency threshold ε and follow

Rule R. If idle, initiate a download whenever $\lambda \le r/c - \varepsilon$.

In Nash equilibrium (NE) the result typically will be inefficient congestion, because an individual player will not recognize the social cost (longer latency times for everyone else) when choosing to initiate a download. Our game has many NE due to the numerous player permutations that yield the same overall outcome, and due to integer constraints on the number of downloads. Fortunately, the NE are clustered and produce outcomes in a limited range.

To compute the range of total NE total profit V^{NE} for our experiment, assume that all players use the threshold ε=0 and assume again that noise is negligible. No player will earn negative profits in NE, since the option is always available to remain idle and earn zero profit. Hence the lower bound on V^{NE} is zero. Appendix A derives the upper bound $V^{MNE} = T(rC\text{-}cS)/S$ from the observation that it should never be possible for another player to enter and earn positive profits. Hence the maximum NE efficiency is $V^{MNE}/V^* = 4(C\text{-}cS/r)/(1+C - cS/r)^2 = 4U^{MNE}/(1+U^{MNE})^2$. For the parameter values used above (T=120, C=6, S=8, c=2 and r=10), the upper bound NE values are U^{MNE} = 4.4 active users (players), λ^{MNE} =3.08 seconds delay, p^{MNE} = 3.85 points per download, N^{MNE} =171.6 downloads, and V^{MNE}= 660.1 points per period, for a maximum efficiency of 60.4%.

The preceding calculations assume that the number of players m is at least U^{MNE}+1, so that congestion can drive profit to zero. If there are fewer players, then in Nash equilibrium everyone is always downloading. In this case there is excess capacity $a = U^{MNE} +1\text{-} m = C+1\text{-} cS/r - m > 0$ and, as shown in the Appendix, the interval of NE total profit shrinks to a single point, $P\, m$=$Tram/S$.

What happens if the background noise is not negligible? As explained in the Appendix, the noise is mean-reverting in continuous time. Thus there will be some good times when effective capacity is above C and some bad times when it is lower. Since the functions V^{MNE} and V^* are convex in C (and bounded below by zero), Jensen's inequality tells us that the loss of profit in bad times does not fully offset the gain in good times. When C and m are sufficiently large (namely, $m > C > cS/r + 1$, where the last expression is 2.6 for the parameters above), this effect is stronger for V^* than for V^{MNE}. In this case Nash equilibrium efficiency V^{MNE}/V^* decreases when there is more noise. Thus the prediction is that aggregate profit should increase but that efficiency should decrease in the noise amplitude $s / \sqrt{2t}$ (see Appendix A.1).[1]

A key testable prediction arises directly from the Nash equilibrium benchmarks. The null hypothesis, call it full rent dissipation, is that when noise amplitude is small, aggregate profits will be V^{MNE} =$Tram/S$ in periods with excess capacity a>0, and will be between 0 and $V^{MNE} = T(rC\text{-}cS)/S$ in periods with no excess capacity. The corresponding expressions for efficiency have already been noted.

One can find theoretical support for alternative hypotheses on both sides of the null. Underdissipation of rent, i.e., aggregate profits higher than V^{MNE}, will arise if players can maintain positive thresholds ε in Rule R. A libertarian justification for this hypothesis is that players somehow self-organize to partially internalize the congestion externality; see e.g. [6]. For example, players may discipline each other using punishment strategies. Presumably the higher profits would emerge in later periods as self-organization matures. A behavioral economics justification is that players have positive regard for the other players' payoffs, and will restrain themselves from going after the last penny of personal profits in order to reduce congestion. One might expect this effect to weaken a bit in later periods.

Overdissipation of rent, i.e., negative aggregate profits, is the other possibility. One theoretical justification is that players respond to relative payoff and see increasing returns to downloading activity; see e.g., [7]. A behavioral economics justification is that people become angry at the greed of other players and are willing to pay the personal cost of punishing them by deliberately increasing congestion; see e.g., [2]. Behavioral noise is another justification. For example, [1] uses quantal response equilibrium, in essence Nash equilibrium with behavioral noise, to explain overdissipation in all-pay auctions.

Further insights may be gained from examining individual decisions. The natural null hypothesis is that human players follow Rule R with idiosyncratic values of the threshold ε. According to this hypothesis, the only significant explanatory variable for the download decision will be $\lambda - r/c = \lambda$ -5 sec, where λ is the average latency currently displayed on the screen. An alternative hypothesis (which occurred to us only after looking at the data) is that some humans best-respond to Rule R behavior, by anticipating when such behavior will increase or decrease λ and reacting to the anticipation.

The experiment originally was motivated by questions concerning the efficiency impact of automated Rule R strategies. The presumption is that bots (and human players in auto mode) will earn higher profits than humans in manual mode.[2] How strong is this effect? On the other hand, does a greater prevalence of bots depress everyone's profit? If so, is the second effect stronger than the first, i.e., are individual profits lower when everyone is in auto mode than when everyone is in manual mode? The simulations reported in [8] confirm the second effect but disconfirm the social dilemma embodied in the last question.

---

[1] The simulations reported in [8] suggest an alternative hypothesis: profits increase in noise amplitude times (s – ½), where s is the fraction of players in auto mode. It should be noted that their bot algorithm supplemented Rule R with a Reload option.

[2] Indeed, a referee of our grant proposal thought it was redundant to use human subjects because the bots obviously would perform better.

Our experiment examines whether human subjects produce similar results.

4. RESULTS

We begin with a qualitative overview of the data. Figure 1 below shows behavior in a fairly typical period. It is not hard to confirm that bots indeed follow the variable l =average delay: their download requests cease when l rises above 4 or 5, and the red line indicating the number of bots downloading stops rising. It begins to decline as existing downloads are completed. Likewise, when l falls below 4 or 5, the bot downloads start to increase.

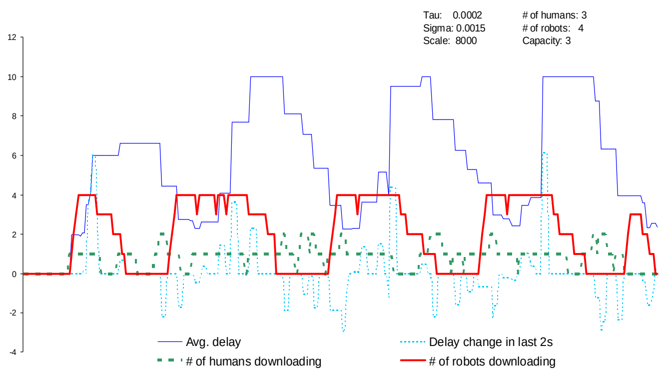

Figure 1: Session 09-12-2002, Period 18

The striking feature of Figure 1 is that the humans are different. They appear to respond as much to the *change in* average delay. Sharp decreases in average delay encourage humans to download. Perhaps they anticipate further decreases, which would indeed be likely if most players use Rule R. We shall soon check this conjecture more systematically.

Figure 2 shows another surprise, strong overdissipation. Both bots and humans lose money overall, especially bots (which include humans in the auto mode). The top half of human players spend only 1% of their time in auto mode, and even the bottom half spend only 4% of their time in auto mode. In manual mode, bottom half human players lose lots of money but at only 1/3 the rate of bots, and top half humans actually make modestly positive profit.

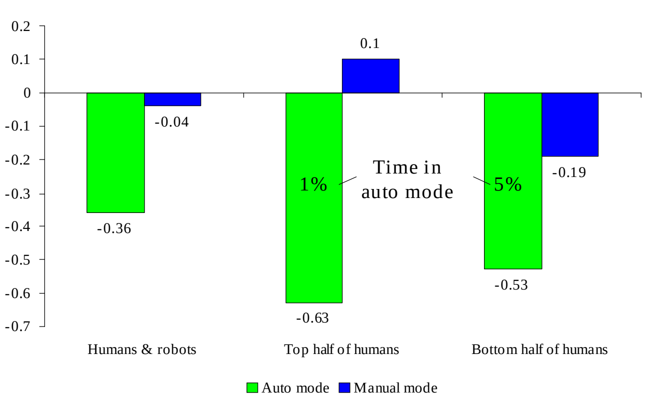

Figure 2. Profit per second in auto and manual mode

Figure 3 offers a more detailed breakdown. When capacity is small, there is only a small gap between social optimum and the upper bound aggregate profit consistent with Nash Equilibrium, so Nash efficiency is high as shown in the green bars for C=2, 3, 4. Bots lose money rapidly in this setting because congestion sets in quickly when capacity is small. Humans lose money when inexperienced. Experienced human players seem to avoid auto mode and learn to anticipate the congestion sufficiently to make positive profits. When capacity is higher (C=6), bots do better even than experienced humans, perhaps because they are better at exploiting the good times with excess capacity. Of course, overdissipation is not feasible with excess capacity: in NE everyone downloads as often as physically possible and everyone earns positive profit.

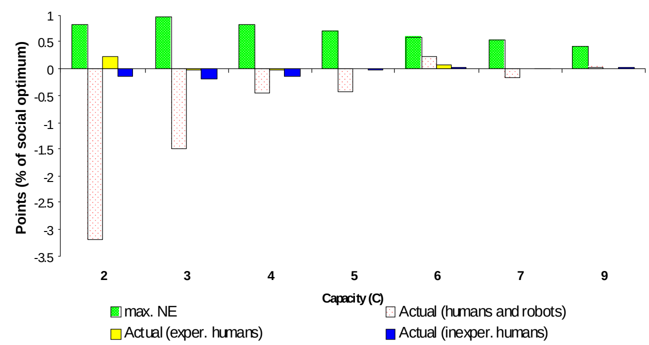

Figure 3. Theoretical and actual profits as percentage of the social optimum

We now turn to more systematic tests of hypotheses. Table 2 reports OLS regression results for profit (net payoff per second) earned by four types of players. The first column shows that bots (lumped together with human players in auto mode) do much better with larger capacity and with higher noise amplitude, consistent with NE predictions. The effects are highly significant, statistically as well as economically. The other columns indicate that humans in manual mode are able to exploit increases in capacity only about half as much as bots, although the effect is still statistically highly significant for all humans and top half of humans. The next row suggests that bots but not humans are able to exploit higher amplitude noise. The last row of coefficient estimates finds that, in our mixed bot-human experiments, the interaction [noise amplitude with excess fraction of players in auto mode] has the opposite effect for bots as in [8], and has no significant effect for humans.

Table 2. OLS Estimates of profit rates

Excess capacity is a=C-m-0.6; Noise is sigma /• (2Tau); and s is fraction of all players in auto mode per period. All coefficients are significant at $p<0.01$, except [a]: $p=0.04$ and [b]: $p=0.06$

For:	Auto mode	Manual mode		
Indep.	All	All	Top	Bottom

variables	players	humans	half	half
Intercept	0.88	0.48	0.64	not sig.
Excess capacity	0.69	0.27	0.29	0.17[a]
Excess capacity^2	0.08	0.03	0.04	not sig.
Noise	0.53	not sig.	not sig.	-0.12[b]
Noise*(s-1/2)	-1.81	not sig.	not sig.	not sig.
NOBS	1676	1222	640	582

Table 3 below reports a fine-grained analysis of download decisions, the dependent variable in the logit regressions. Consistent with Rule R (hardwired into their algorithm), the bots respond strongly and negatively to the average delay observed on the screen minus $r/c = 5$. Surprisingly, the regression also indicates that bots are more likely to download when the observed delay increased over the last 2 seconds; we interpret this as an artifact of the cyclical congestion patterns. Fortunately the delay coefficient estimate is unaffected by omitting the variable for change in delay.

Human players react in the opposite direction to delay changes. The regressions confirm the impression gleaned from Figure 1 that humans are much more inclined to initiate download requests when the observed delay is decreasing. Perhaps surprisingly, experienced humans are somewhat more inclined to download when the observed delay is large. A possible explanation is that they then anticipate less congestion from bots.

Table 3: Logit regressions

All coefficients are significant at $p<0.01$, except [a]: $p=0.047$

For:	Auto mode		Manual mode		
Indep. Variables	All players		All humans	Top half	Bottom half
Intercept	-2.08	-2.12	-2.51	-2.31	-2.74
Avg. delay -5s	-0.31	-0.31	0.03	0.06	-0.01[a]
2s change in avg. delay	0.08		-0.19	-0.30	-0.08
NOBS	163602	163602	186075	92663	93412

5. DISCUSSION

The most surprising result is that human players outperform the current generation of automated players (bots). The bots do quite badly when capacity is low. Their decision rule fails to anticipate the impact of other bots and neglects the difference between observed congestion (for recently completed download attempts) and anticipated congestion (for the current download attempt). Human players are slower and less able to exploit excess capacity (including transient episodes due to random noise), but some of them are far better at anticipating and exploiting congestion trends created by bots. In our experiment the second effect outweighs the first, so humans earn higher profits overall than bots.

Perhaps the most important questions in our investigation concerned rent dissipation. Would human players find some way to reduce congestion costs and move towards the social optimum, or would they perhaps create even more congestion than in Nash equilibrium? Sadly, overdissipation outcomes are most prevalent in our data.

The Nash comparative statics, on the other hand, generally help explain the laboratory data. Nash equilibrium profit increases in capacity and noise amplitude, and so do observed profits.

Several directions for future research suggest themselves. First, one might want to look at smarter robots. Preliminary results by Pommerenke (2003) show that it is not as easy as we thought to find more profitable algorithms for robots; linear extrapolation from available data seems rather ineffective. Second, one might want to probe the robustness of the overdissipation result. It is surely worth checking in humans-only and in bots-only environments, and for larger numbers of players. One could also check alternative congestion functions to the mean-reverting noisy M/M/1 queuing process. More generally, one could investigate mechanisms such as congestion taxes to see whether they enable humans and robots in congestible real-time environments to increase profits above the Nash equilibrium level.

6. ACKNOWLEDGMENTS

The National Science Foundation funded this work under grant IIS-9986651. We are also grateful for the research assistance of Alessandra Cassar, Garrett Milam, Ralf Hepp, and Kai Pommerenke, and the programming support of Neil Macneale, Jeremy Avnet and Nitai Farmer. We benefited from comments by colleagues including Joshua Aizenman, Eileen Brooks, Rajan Lukose, Nirvikar Singh, and Donald Wittman, and by HKUST conference participants, especially Yan Chen, Rachel Croson and Eric Johnson.

7. REFERENCES

[1] Anderson, S., Goeree, J and Holt, C. The All-Pay Auction: Equilibrium with Bounded Rationality. *Journal of Political Economy, 106*, 4 (August 1998), 828-853.

[2] Cox, J. and Friedman, D. A tractable model of reciprocity and fairness. UCSC Economics Working Paper 546 (October 2002)

[3] *Economist* magazine. Robo-traders. Nov 30, 2002, p65.

[4] Feller, W. *An Introduction to Probability Theory and Its Applications, Vol 2.* NY: Wiley. 1968.

[5] Friedman, E., Shor M., Schenker, S. and Sopher, B. An Experiment on Learning with Limited Information: Nonconvergence, Experimentation Cascades, and the Advantage of Being Slow. *Games and Economic Behavior* (forthcoming)

[6] Gardner, R., Ostrom, E. and Walker, J. Covenants With and Without a Sword: Self-Governance is Possible. *American Political Science Review* 86, 2 (June 1992), 404-417.

[7] Hehenkamp, B., Leininger, W., and Possajennikov, A. Evolutionary Rent Seeking. *CESifo Working Paper* 620, December 2001.

[8] Maurer, S. and Huberman, B. Restart Strategies and Internet Congestion. *Journal of Economic Dynamics & Control* 25 (2001) 641-654.

[9] Ochs, J. The Coordination Problem in Decentralized Markets: An Experiment. *The Quarterly Journal of Economics* 105, 2. (May, 1990) 545-559.

[10] Rapoport, A., Seale, D., Erev, I., & Sundali, J. Equilibrium play in large market entry games. *Management Science* 44 (1998) 119-141.

8. APPENDIX A: TECHNICAL DETAILS

8.1 Latency and Noise. Following the noisy M/M/1 queuing model of [8], latency for a download request initiated at time t is $\lambda(t) = \frac{S[1+e(t)]_+}{[1+C-U(t)]}$ if the denominator is positive, and otherwise is $\lambda^{max}>0$. Here $[x]_+ = \max\{x, 0\}$, while C is the capacity chosen for that period, S is a scale constant, and $U(t)$ is usage, the number of downloads initiated but not yet completed at time t. The experiment truncates actual latency to [0.2, 10.0] seconds; the lower truncation earns the 10 point reward but the upper truncation at λ^{max} =10 seconds does not.

The random noise $e(t)$ is Normally distributed with volatility σ and unconditional mean 0. The noise is mean reverting in continuous time and follows the Ornstein-Uhlenbeck process with persistence parameter $\tau>0$; see e.g., [4, pg 336]. That is, $e(0) = 0$ and, given the previous value $x = e(t-h)$ drawn at time $t-h>0$, the algorithm draws a unit Normal random variate z and sets

$$e(t) = x\exp(-\tau h) + z\sigma\sqrt{[1-\exp(-2\tau h)]/(2\tau)}\ .$$

In the no-persistence limit $\tau \longrightarrow 0$ (.e., no mean reversion) we have Brownian motion with conditional variance $\sigma^2 h$, and $e(t) = x + z\sigma\sqrt{h}$. In the long run limit as $h \longrightarrow \infty$ we recover the unconditional variance $\sigma^2/(2\tau)$. The appropriate measure of noise amplitude in our setting therefore is its square root $\sigma/\sqrt{2\tau}$.

In our experiments we used two levels each for σ and τ. Rescaling time in seconds instead of milliseconds, the levels are 2.5 and 1.5 for σ, and 0.2 and 0.02 for τ. Figure A1 shows typical realizations of the noise factor $[1+e(t)]_+$ for the two combinations used most frequently, low amplitude (low σ, high τ) and high amplitude (high σ, low τ).

A.2. Efficiency, no noise case. Social value V is the average net benefit $\pi = r - \lambda c$ per download times the total number of downloads $n \approx UT/\lambda$, where λ is the average latency, T is the length of a period and U is the average number of users attempting to download. (The approximation is due to integer constraints in the identity $n\lambda=UT$=active player-seconds.) Assume that $\sigma = 0$ (noise amplitude is zero) so average latency $\lambda = S/(1+C-U)$, and that the last approximation is exact. Then the first order condition for maximizing V yields $U^* = 0.5(1+C - cS/r)$ and $\lambda^* = 2S/(1+C+cS/r)$, and so maximized social value is $V^*=0.25S^{-1}Tr(1+C - cS/r)^2$.

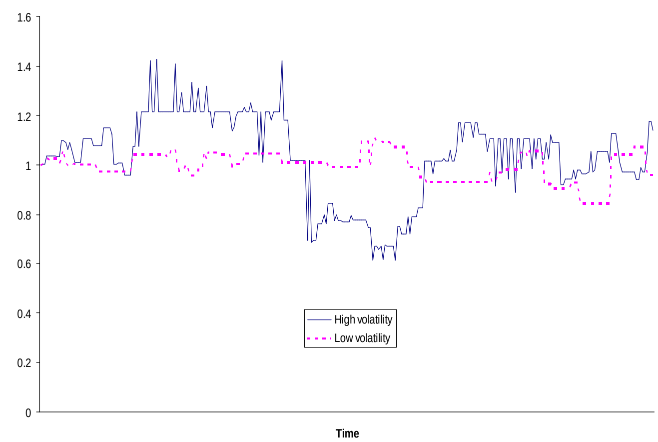

Figure A1. Noise in session 10/3/03, periods 1 and 4

To obtain the upper bound on social value consistent with Nash equilibrium, suppose that more than 10 seconds remain, the player currently is idle and the expected latency for the current download is λ. The zero-profit latency is derived from $0 = \pi = r - \lambda c$. Now $\lambda = r/c$ and (from the earlier expression for average latency) we see that the associated number of users is $U^{**} = C + 1 - cS/r = 2U^*$. Hence the minimum number of users consistent with NE is $U^{MNE} = U^{**} - 1 = C - cS/r$. The associated latency is $\lambda^{MNE} = rS/(r+cS)$, and the associated profit per download is $\pi^{MNE} = r^2/(r+cS)$, independent of C. The maximum number of downloads is $N^{MNE} = TU^{MNE}/\lambda^{MNE} = T(r+cS)(rC-cS)/(r^2S)$. Hence the upper bound on NE total profit is $V^{MNE} = N^{MNE}\pi^{MNE} = T(rC-cS)/S$, and the maximum NE efficiency is $V^{MNE}/V^* = (C-cS/r)/(1+C - cS/r)^2 = 4U^{MNE}/(1+U^{MNE})^2 \bullet Y$. Since $dU^{MNE}/dC = 1$, it follows that $dY/dC < 0$ iff $dY/dU^{MNE} < 0$ iff $1 < U^{MNE} = C - cS/r$. It is also easy to verify that Y is $0(1/C)$.

A.3 Bot algorithm. In brief, the bot algorithm uses Rule R with a random threshold ε drawn independently from the uniform distribution on [0, 1.0] sec. The value of λ is the mean reported in the histogram window, i.e., the average for download requests completed in the last 10 seconds. Between download attempts the algorithm waits a random time drawn independently from the uniform distribution on [.25, .75] sec.