Traffic Prediction on the Internet*

Y. Baryshnikov†, E. Coffman‡, D. Rubenstein‡, and B. Yimwadsana‡

† Bell Labs, Lucent Technologies
Murray Hill, NJ 07974

‡ Department of Electrical Engineering
Columbia University
New York, NY 10027

Paper #185, 6 pages

Abstract— **Large swings in the demand for content are commonplace within the Internet. To service this demand adequately without over-provisioning requires that the number of servers processing requests be adjusted dynamically with the demand. Since such adjustments require negotiation and the transfer of data across a distributed network, there is a time lag between the decision to adjust and the availability of increased capacity. Failure to perform such adjustments sufficiently in advance can lead to service degradation or even denial of service for a large number of requests. Hence, with enough advance notice, knowledge of future traffic levels can lead to significant improvements in service quality.**

Performing accurate predictions appears to be a daunting challenge at first glance, but this paper shows that, when applied to Internet traffic, even elementary prediction techniques can have a surprising forecasting power. Our analysis of several different Internet traces reveals that server overloads can often be seen well in advance. This allows steps to be taken to reduce substantially the degradation of service quality. We discuss various systems, such as server and peer-to-peer cooperatives, where traffic predictors (even imperfect ones) can be used to great advantage in reducing latencies and improving resilience to hotspots, whether they be caused by flash crowds, router breakdowns, link outages, . . ., and whether they be innocent in origin or produced by vandals or terrorists.

I. Introduction

Traffic congestion in many settings, such as the automobile and airline transportation networks, is often predictable in the period just before it occurs, and far enough in advance that something can be done to mitigate its effects. A thesis of this paper is that this observation also applies to the Internet. Even moderately accurate predictions can be used to good effect in driving the migration of files/objects in the overlay networks of peer-to-peer server/cache cooperatives. The results presented here also show that there are simple, highly successful algorithms for predicting many Internet "hotspots," the intolerable surges of traffic that occasionally bring web servers and routers to a virtual standstill. A colorful, more recent term is "flash crowds;" other terms include the "slash dot effect," "storms," and "floods."

Papers on hotspot properties generally refer to their spontaneity, their short lifetimes, and their instant, sudden, or nearly discontinuous ramp up of traffic. But as noted in [JKR02], this last property is only valid on large time scales, much larger than the time scale needed to react to an approaching hotspot. That a hotspot is visible in the distance can be explained in a number of ways. For example, there is the simple phenomenon of "passing the word" on a "hot" document. This process grows exponentially fast, but this in itself does not rule out adequate advance warning, as the initial growth is approximately linear. While the performance of our predictors is very good, as seen in Section V, it is worth noting that even a moderate to poor predictor can offer substantial cost savings. For, in analogy with hurricane forecasting, the over-all cost of a missed hotspot far exceeds the aggregate cost of false alarms (preparing for hotspots that do not materialize).

II. Related Work

Methods for alleviating the effects of hotspots are discussed at some length in the report [CJNR01], which is the basis of a grant[1] for research into all aspects of the hotspot problem. More recently, Jung et al [JKR02] study hotspot properties determined by traffic patterns, client characteristics, and file-referencing characteristics. They introduce techniques that content-delivery networks can employ to adapt to large increases in load. Thus, their ap-

This research was supported in part by NSF grant ANI-0117738

proach is reactive and assumes that all provisioning of servers can be done in advance; in contrast, ours is proactive: we introduce techniques for hotspot prediction and measures that can be taken well in advance of an approaching hotspot.

Several proposed schemes would benefit from an improved ability to forecast upcoming traffic loads. The Oceano project at IBM [Res] provides a “farm” of additional server resources to customers that can be used to service customer demand during periods of overload. However, there is no solution proposed for predicting when such services are necessary. Alternatively, Stading et al [SMB02] describe a peer-to-peer directed-search caching scheme for load balancing in heavy load conditions, once again an essentially reactive approach. To prevent overloading, participants push content to other participants when the request rate for that content exceeds a predefined threshold. It is similar in purpose to the ”mutual aid society for hotspots” proposed in [CJNR01]. The system in [SMB02] is based on a distributed hash table overlay.

III. Prediction algorithm

In the traces studied in the next section, a traffic observation, r_t, gives the number of page requests in time unit $(t-1, t]$. A typical time unit is one second, but depending on the structure of data, prediction algorithm, and requested reaction time, the data may be accumulated into time slots of different length. *In the following, all parameters are to be taken as integer-valued, unless stated otherwise.*

The prediction of Internet traffic (in particular, the Web server request flow) should be considered within the general framework of time series analysis, filtering, and prediction. While there exists an extended and rich theory servicing applications in such diverse areas as signal processing and econometrics, one would in fact be quite surprised to see that “standard” methods would suffice in the context of Internet traffic, which is notorious for its burstiness.

Even so, we decided to start our investigation by testing what is arguably the simplest reasonable traffic prediction algorithm: linear extrapolation. As it turns out, in a number of important cases, it performs remarkably well.

The structure of prediction algorithms is as follows. The defining parameters of any prediction algorithm are the duration, $W_p \geq 0$, of a *prediction window* and a time, $\tau \geq 0$, called the *advance notice*. A prediction at time t is a mapping from the observations $r : W_p \to \mathbf{R}_+$ in the prediction window $[t - W_p, t]$ to a number $p_t \geq 0$ of requests predicted to appear in the interval $[t+\tau, t+\tau+1]$, τ time units in the future.

The Linear Fit (LF) algorithm uses a mapping by which p_t extrapolates a least-squares linear fit, i.e., $p_t = f_t(t + \tau)$, where $f_t(s) = a_t s + b_t$ is a linear function minimizing $\sum_{i=t-W_p}^{t} [f_t(i) - r_i]^2$, the mean quadratic deviation on the window $[t - W_p, t]$.

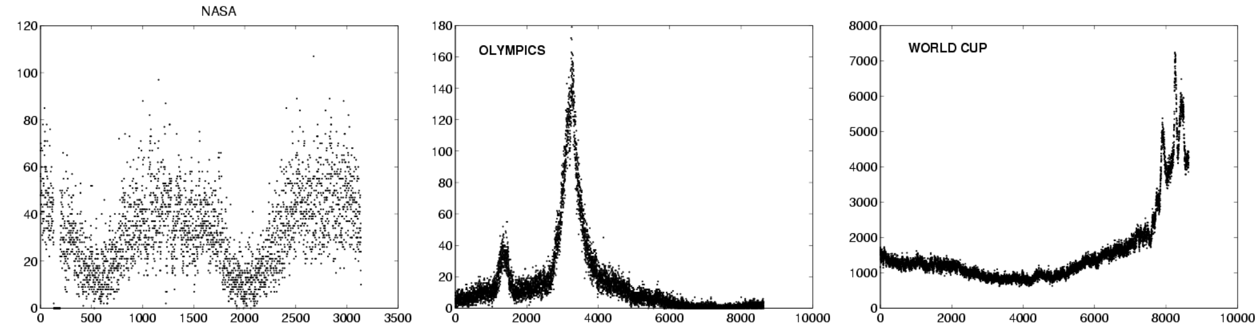
The simplicity of LF leads to a very straightforward realization with a virtually unbeatable performance: it is $O(W + T)$ where W is the window size and T is the uptime duration.

The simplicity is also responsible for certain potential limitations. It is rather clear that linear predictors will have some drawbacks if the prediction window size is poorly adjusted to the parameters of burstiness governing the request flow. Thus, if the flow is prone to step-like bursts of activity, the linear predictor will tend to overestimate the flow at the beginning of the step; similarly, the linear predictor will underestimate the flow at the early stages of a sudden linear increase of activity. More generally, the structure of the power spectrum of the time series describing the requests flow influences strongly the quality of a linear predictor.

Note that the LF algorithm can easily be extended to predict not only the level of traffic at some future time, τ, but also the time at which the traffic level will exceed a given threshold request rate, H. If the linear fit has a positive slope, the algorithm simply computes the time at which the curve crosses the threshold.

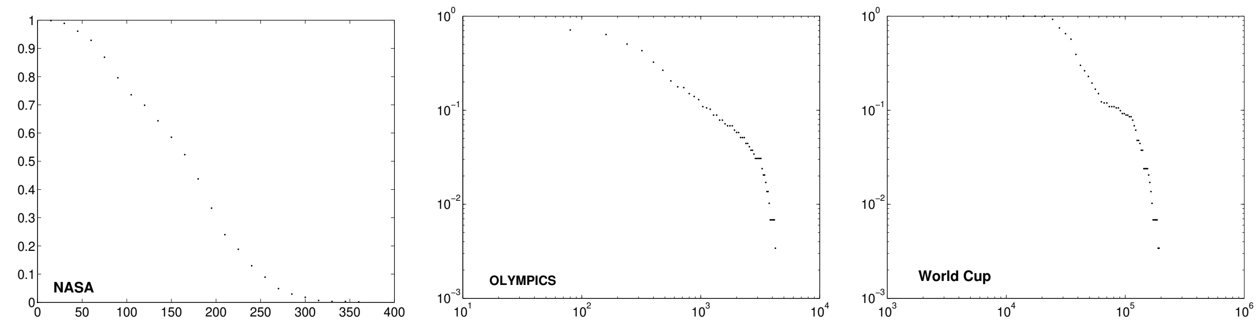
A natural next step after LF is to look at extrapolation polynomials of degree higher than one. Here, we again choose the coefficients of the polynomial using the least squares approximation procedure, and then proceed to evaluate the resulting polynomial at the future time, τ units ahead. The complexity of such algorithms remains low, $O(W + D^3T)$ (where D is the degree of the polynomial, and 3 in the exponent comes from the $D \times D$-matrix inversion, and can be readily reduced for large D, if needed).

One complication which arises in this approach is



(a) NASA (b) Olympics (c) World Cup

Fig. 1. Fragments of three sample traces



(a) NASA (b) Olympics (c) World Cup

Fig. 2. Tail of the distribution for the number of page requests per time slot.

that the polynomial need not be monotonic, so that the first instance of the threshold crossing might be earlier than $t+\tau$ (t here is the current time), while the value of the extrapolated traffic at $t+\tau$ can be below threshold. Another difficulty, arising even for LF is that there exists a possibility that the extrapolated function crosses the threshold even *within* the window of observation, where there is no factual threshold crossing. As it turns out, these phenomena are rare on the traces we considered, and had no significant impact on the quality of predictions.

More sophisticated prediction algorithms incorporate filtering. Such algorithms will ignore brief inversions of an otherwise well established trend, and will weight recent observations more than old ones.

IV. Characterizing the request flow

As the test objects of this study, we have chosen several activity logs for various web servers. We tried to diversify the phenotypes of the traces to probe for the limits of applicability of our algorithms.

It is well known (see e.g. [JKR02], [ISZ99]) that a load-based taxonomy of web server traffic is both an extremely important and an extremely challenging problem. The behavior of the traffic is shaped by a combination of so many technological, sociological, psychological (to name a few) factors that a quantification of even basic patterns reflecting this behavior would be a major breakthrough. We intend to return to this topic elsewhere, and restrict our attention in this study to an intermediate or "mesoscopic" scale of traffic activity, in intervals from 10 to 10^3 seconds, that is, on a time scale relevant to proactive control of CDN's. (For traffic predictions on a macroscopic scale, e.g., months to years, see for example [BH97].)

As an example, in Figure 1 we show fragments of graphs of web server loads in three cases: A NASA server (August 1995); a Nagano Winter Olympics server(1998) and Soccer World Cup server (1998). Time increases along the x-axis, and the y-axis plots the number of requests in successive 10-second slots. One sees immediately that these graphs are strikingly different; as we will see, the predictive powers of our algorithms when applied to these traces also vary dramatically. Figure 2 shows the tails of distributions of requests per time slot. Here, the number of requests that can occur within 10-second time slots is varied along the x-axis and the y-axis plots the probability that the number of requests in a slot exceeds this value. Note the heavy-tail properties of the Olympics and World Cup distributions, which are given in log-log plots (the plot for the Olympics data confirms the log-normal estimate of [ISZ99]). By contrast, the distribution for the NASA data is relatively flat.

As is common in prediction problems, the performance of the prediction algorithms strongly depends on the power spectrum (see [GS92] for example), that is, the absolute value of the Fourier transform of the time series. Figure 3 plots the autocorrelation functions (inverse Fourier transforms) of the power spectra for a sample of the three data sets we have studied. The x-axis is the time parameter of the autocorrelation function given on the y-axis.

As one can see, these three autocorrelation functions are quite different. Typically, more rapidly falling autocorrelation functions are more "jittery" and more difficult from the predictive point of view.[1]

One would clearly expect that, on the time scale of $5 - 30$ minutes, the performance of any linear prediction algorithm like LF would be optimal for the World Cup data, somewhat worse for the Olympic data and worse yet for the NASA traces. As it turns out, this intuition is exactly right.

[1]One can in fact read a wealth of information off the power spectra of the request flow. It is not surprising to find strong peaks at frequencies reciprocal to $86400s$ – the duration of a day. Some other peaks were more difficult to interpret, like those at about $10^{-3}s^{-1}$. It seems likely that they are caused mainly by browsers automatically refreshing a web page. However, we must leave this interesting line of study for another occasion, as it deviates from our main topic.

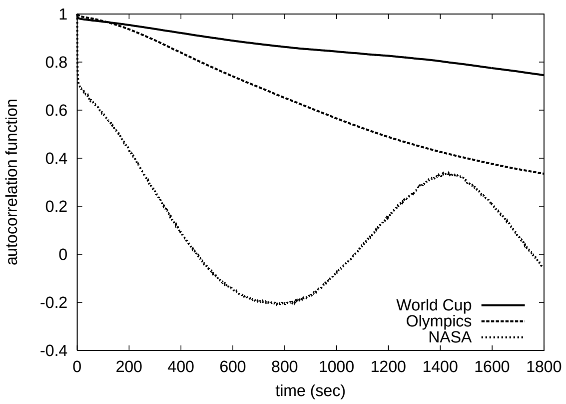


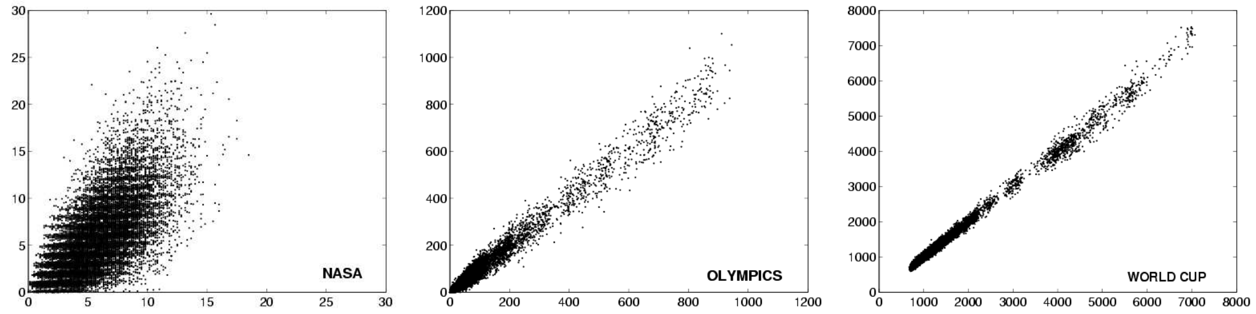
Fig. 3. Autocorrelation functions for the World Cup, Olympics, and NASA traces.

V. Performance

As shown in Figure 4, plots of $(p_t, r_{t+\tau}),\ t \geq 1$, give a visual estimate of forecasting power. A point plotted at (x, y) represents a slot containing x requests for which the predicted value was y requests. The quality of the prediction is high when the points tend to be close to the diagonal. If the predictor were ideal, the points would then follow the diagonal precisely. Behavior in the worst case would show that reality is independent of the prediction, and the patterns, if any, in the scatter plot would have no relation to the diagonal.

We see a striking correlation of the performance of the linear fit algorithm with the properties of the autocorrelation function. Several more scatter plots were created for the quadratic extrapolation on the same set of data. These plots were very similar, so in the interests of conserving space, they have been omitted.

Figure 5 tests the comparative quality of the linear and quadratic predictive powers. As a measure of comparison, beyond the visually satisfying but non-quantifiable scatter plots, we decided to use the average of the relative prediction errors, $\left|\frac{p_i - r_i}{r_i}\right|$, where p_i and r_i are corresponding predicted and "real" values; this is the y-axis in Figure 5. We used only the World Cup and the Olympic data, where both algorithms show significant predictive power, and let the size of the observation window, the x axis in Figure 5, be variable. The average relative error is computed only for sufficiently large actual data where prediction performance is important. The lower lim-



(a) NASA (b) Olympics (c) World Cup

Fig. 4. Scatter plots contrasting real versus predicted values within the three traces.

its were 60 and 4000 for the Olympics and World Cup data respectively. As was to be expected, neither algorithm is an all-discipline winner. While the superiority of LF goes against intuition perhaps (since the quadratic approximation to the window data is by definition at least as good as the linear one), one can argue that the actual goal is not interpolating the data, but extrapolating it, and so the quadratic algorithm might tend to overreact to the oscillations on a finer scale. These measurements can also be used for fine tuning of the prediction algorithms, by changing the window size. For example, the figures show that, in many cases, there is an optimal window size W_{opt}; as the window decreases below W_{opt}, it loses useful data that should influence the prediction, and as it increases beyond W_{opt}, it introduces superfluous data that should not influence the prediction.

VI. Implications and Uses

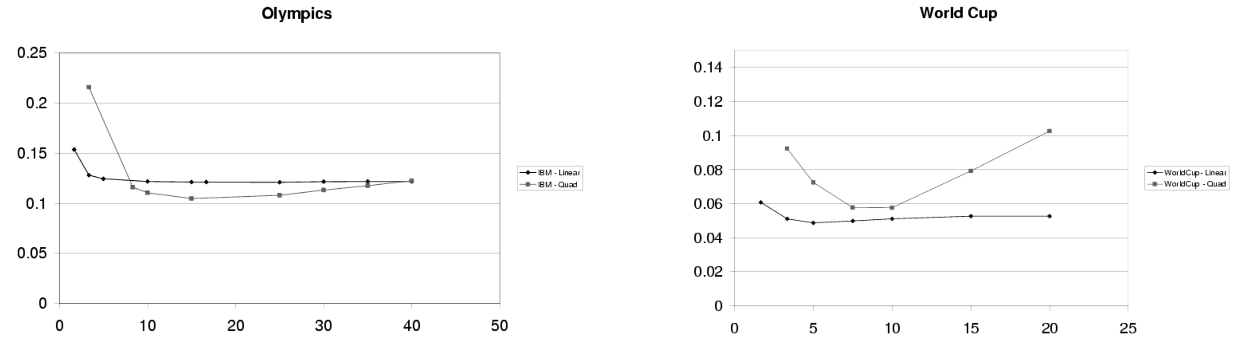
In this section, we touch briefly on the uses of our results in real content serving systems. The objective is a system that can adjust dynamically to an unexpected surge in request load, and with enough advance notice that the adjustments can be made before traffic conditions reach the point where the server, without assistance, would become overloaded. The bandwidth and memory required by the transfer of an object to another server is one reason why advance notice is needed; our experiments suggest that the needed lead time does indeed exist. Morover, the requirements for long lead times can be reduced substantially by appropriately replicating objects elsewhere in the network.

While it may not be necessary to complete an object's transfer prior to its demand spike, it is necessary to update the system so that users can be turned directly to the alternative serving points. This process itself may take several seconds, depending on the type of system. For example, a delivery system that utilizes the Domain Name System (DNS) must update numerous servers throughout the network. This is because each DNS independently caches its own set of entries that determine where requests for content should be forwarded. One possible solution for a DNS model that requires forecasting capabilities is to add a mechanism to the DNS system that can push modifications from servers that have been updated to those that need updating.

Peer-to-peer content delivery systems can exploit an ability to forecast overloads. For instance, directed search, peer-to-peer approaches such as those described in [RFH$^+$01], [RD01], [SMK$^+$01] rely on cached copies of "hot" objects at various points within the P2P overlay to protect the host nodes from overload. A simple solution for such systems is to return and store a copy along all points that were used to forward the request. However, given the Zipf-like nature of the distribution of objects requested in web systems [BCF$^+$99], caching every requested object in this manner can place an excessive demand on bandwidth and storage at the P2P nodes. These demands can easily be reduced by caching in this manner only those objects whose request forecast indicates high levels of activity in the near future.

VII. Conclusion

We have shown that even very simple prediction algorithms have a significant predictive power. One



(a) Olympics

(b) World Cup

Fig. 5. Relative error of predicted versus actual request rate.

meta-conclusion is that they perform extremely well in situations where they are needed (that is when hot spots are likely to occur), and much worse in situations where there is no actual need for hotspot prediction. This is not only the best of the scenarios available to us, it is plainly a good one!

There is much that can be done to further develop effective prediction of Internet traffic. First, we intend to examine several more traces, currently in hand, and we are making an effort to acquire appropriate 9/11 data. Second, additional experiments with varying window sizes, and advance notices are needed in order to better understand the functional dependence of predictive power on these parameters. Third, we would like to design in detail and "field test" prediction mechanisms.

REFERENCES

[BCF+99] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proceedings of IEEE INFOCOM*, New York, NY, April 1999.

[BH97] J. Bolot and P. Hoschka. Performance Engineering of the World Wide Web: Application to Dimensioning and Cache Design. *World Wide Web Journal*, pages 185–195, 1997.

[CJNR01] E. Coffman, P. Jelenkovic, J. Nieh, and D. Rubenstein. The Columbia HotSpot Rescue Service. Technical report, Columbia Univeristy Technical Report, February 2001.

[GS92] G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford University Press, New York, NY, 1992.

[ISZ99] A. Iyengar, M. Squillante, and L. Zhang. Analysis and Characterization of Large-Scale Web Server Acess Patterns and Performance. *World Wide Web*, 2, June 1999.

[JKR02] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In *Proceedings of WWW-2002*, Honolulu, Hawaii, May 2002.

[RD01] A. Rowstron and P. Druschel. Storage Management and Caching in PAST, A Large-scale, Persistent Peer-to-peer Storage Utility. In *Proceedings of ACM SOSP'01*, Banff, Canada, October 2001.

[Res] IBM Research. The Oceano Project. http://www.research.ibm.com/oceanoproject.

[RFH+01] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proceedings of ACM SIGCOMM'01*, San Diego, CA, August 2001.

[SMB02] Tyron Stading, Petros Maniatis, and Mary Baker. Peer-to-Peer Caching Schemes to Address Flash Crowds. In *1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, Cambridge, MA, March 2002.

[SMK+01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *Proceedings of ACM SIGCOMM'01*, San Diego, CA, August 2001.