Understanding Web Performance

By Peter Sevcik and John Bartlett
October 2001

Is there an Internet performance problem? There are two views: The conventional wisdom held by 'Net insiders is that things are just fine. They point to the massive investment in bandwidth that has eliminated many congestion problems, and to the performance of the Keynote Business-40 Index, which has fallen from more than 12 seconds to less than 3 seconds in only four years.

The other view is held by the vast majority of Internet users, who complain that the web is an awfully slow way to do anything useful. To be sure, most connections are slow; no matter whose numbers you use, anywhere from 78 percent to 93 percent of the users in the U.S. use a dial-up modem to connect to the Internet. And most of these dial-up connections actually operate at speeds of about 30 Kbps.

Furthermore, most real users and network shoppers don't visit the Keynote Business-40 sites. Instead, they visit sites like MSN, AOL, Amazon, ICQ, ESPN and Disney, which are designed to be "interesting and cool" rather than optimized for performance; these sites are slow by design. In short, geography, demographics and interest all play important roles in determining what the "Internet experience" will be like.

The Gomez report, "Performance Metrics in Context" (www.gomez.com), describes these effects: Only 47 percent of the e-commerce users surveyed are satisfied with the speed of the web.

Indeed, speed is always among the top five reasons for selecting an on-line service for a business transaction or abandoning a shopping cart. But the need for speed is relative to the type of site being visited (brokerage, shopping, travel) and the type of function being performed (browsing, buying, getting a confirmation).

These findings aren't limited to the consumer market. Since 1997, analyst and author Rebecca Wetzel (rwetzel@rwetzel.com) has been surveying enterprises about what they think of their Internet service provider. Her latest survey shows that while performance is the second highest attribute out of 12 (after reliability) they desire from an access provider, it is third from the *bottom* in the service satisfaction rating.

There is a wide range of acceptable response times depending on the activity, its criticality, etc. But all points along the range of acceptable speeds are trending down. Yesterday's "fast" is considered "slow" today.

NetForecast Report 5055

This is an expanded version of an article with the same title first published in *Business Communications Review*, October 2001. See www.bcr.com.

How Applications Work on a Network

The World Wide Web is a complex system of services that operates on top of the Internet, a separate complex system of connectivity and transport. But despite all the complexity, the interaction and behavior of the transactions between a client (browser) and server (website) is very consistent. Page-load time – from the click on a URL to the point at which the page is completely displayed on the destination PC – is a process that can be boiled down to two functions: discovery and transfer.

Discovery

A user starts the process by instructing his/her browser to open a connection to a destination known by a Universal Record Locator – a URL. URLs are convenient names or handles people give to piles of information or some specific data. The browser must first ask a local Domain Name Server (DNS) to resolve the URL name into an Internet Protocol (IP) address. After the DNS system replies with a specific IP address for the URL, the browser opens a Transmission Control Protocol (TCP) connection to that address.

The process by which the connection is opened is called a three-way handshake -- three packets of information and sequence numbers are exchanged. Once the connection is established, the browser sends a HTTP 'Get' command, asking for the content of the URL. The server replies with a base page, which is a description of what the web page will look like when loaded on the screen along with a list of elements (more URLs) to fill the screen. Then the browser sends a 'Get' for each element, one at a time.

So far, the browser has been “discovering” where to go to get the content and how to proceed on actually calling for the content. Each of the exchanges described above requires processing by the client (desktop PC) and one or more servers on the Internet. Each exchange also requires that at least one packet go from the client to a server with at least one packet coming back in reply. Getting a web page is truly a process of discovery, which has just begun.

Conventions and specifications on how the web operates force a few more discoveries to occur. Each element that must be retrieved from a website requires a separate TCP connection along with, potentially, a new DNS address resolution process. (The attribute “persistent TCP” in HTTP version 1.1 reduces the need for some of the TCP opens, but its effectiveness is limited and sometimes counterproductive.) It is very likely that the browser will be told to get some of the content from other servers that are not even associated with the base web page, such a banner ads. In a typical scenario, the website sends the browser to DoubleClick, which will exchange cookie data to figure out which ad this user should see at this time, and then a URL is sent directing the browser to yet another server to receive the ad.

All of the exchanges described above can be grouped into a number we call “turns.” A turn is a non-content carrying exchange of packets between client and server that requires a round-trip over the network. More specifically, it is a count of each time communications changes direction among these discovery packets. TCP-level acknowledgements (ACKs) are not counted as turns. A turn is a limited to the ping-pong packets that do not move any user visible content.

Think of the number of times you have to swing your head back and forth if you are watching a tennis match from a seat near the net. Now think of the number of times the objective – scoring a point – occurred. The ratio of head turns to total points in the match may make for an interesting game, but they are a sure indication of how long the game will take. Some tennis matches take hours to end! Some websites take long to load for the same reason.

Turns add up. Turns take time. They are a direct byproduct of the quest to make the web simple to build and highly scalable.

Transfer

Once the browser finishes the discovery process for each element, it starts the transfer process of moving the content (text, graphic, photograph, etc.) to the desktop. The transfer is performed by TCP using standard windowing and acknowledgement procedures coded into the client and server operating systems.

TCP is a transfer protocol that is controlled by the receiver. Since the overwhelming ratio of content to be moved is from server to client, it is the client that governs how fast things will move. The client advertises a window size in bytes that it is prepared to receive from the server. Once some or all of the window is successfully received, the client acknowledges and updates the window with a new byte count. If all goes well, the content never stops arriving. The client acknowledges fast enough and the server keeps the connection or "pipe" full. In theory, the transfer should operate at the speed of the slowest link in the system less the overhead of protocol headers.

But things hardly ever go that well. There are often delays in updating the window. The server often waits to get an acknowledgement. If a packet is lost, a retransmission has to occur. TCP also uses a mechanism called "slow-start" to help manage congestion. Since most web elements are small enough to fit into one or two packets, the system is always operating in the start-up (slow) phase of the cycle.

The bottom line is that transfer takes time and the throughput is not nearly that of the slowest link in the system.

Application Profiles

Any transactional application – the web is clearly transactional – can be characterized by payload size and turn count. This report does not cover non-transactional applications like voice and video. We call the payload and turn data an "application profile". Fundamental performance over a network can be derived from only these two numbers.

Figure 1 shows the wide range of payload and turns from our library of more than fifty applications. Each circle in Figure 1 encompasses the profiles of the most common user tasks for each application. They are grouped by major application genre.

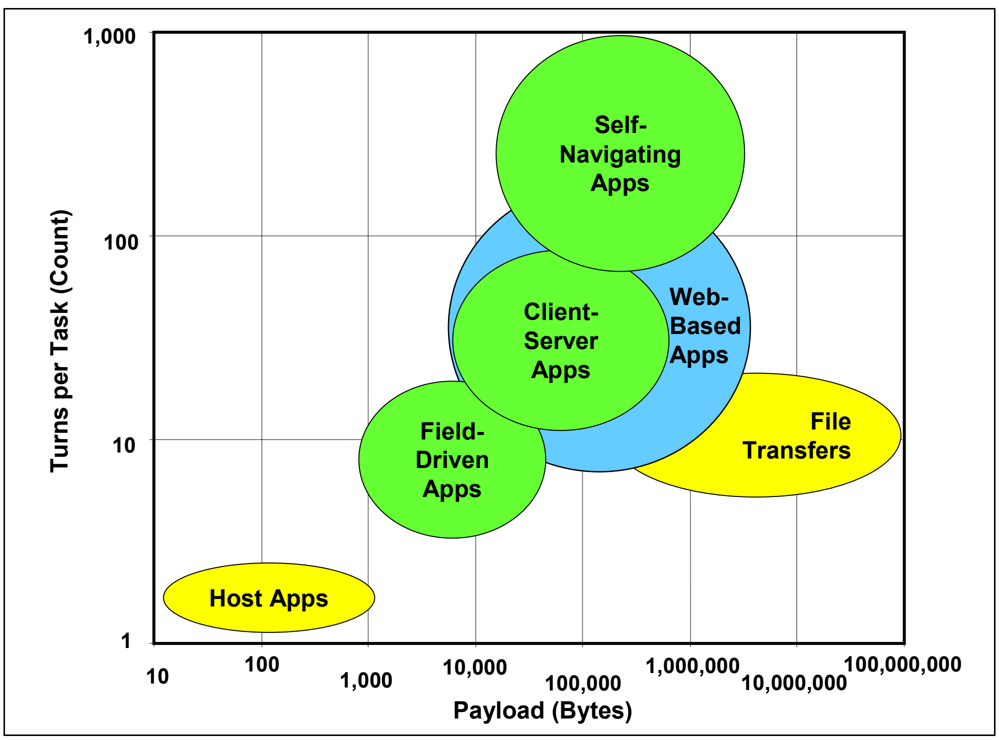

Figure 1 – Wide Range of Application Profiles

Our first significant study of web traffic and its application profile was performed in 1995, when we watched 20,000 users at a single large company attack the web with gusto. In 1995, the typical business web home page had a profile of 50,000 bytes payload and 20 turns. Using the Keynote Business-40 (KB40) as a representative sample of business sites, the current average KB40 profile is 115,000 bytes payload and 40 turns. Figure 2 shows the application profile of each site in the KB40 along with the average composite profiles for three summers where we gathered significant data on business sites.

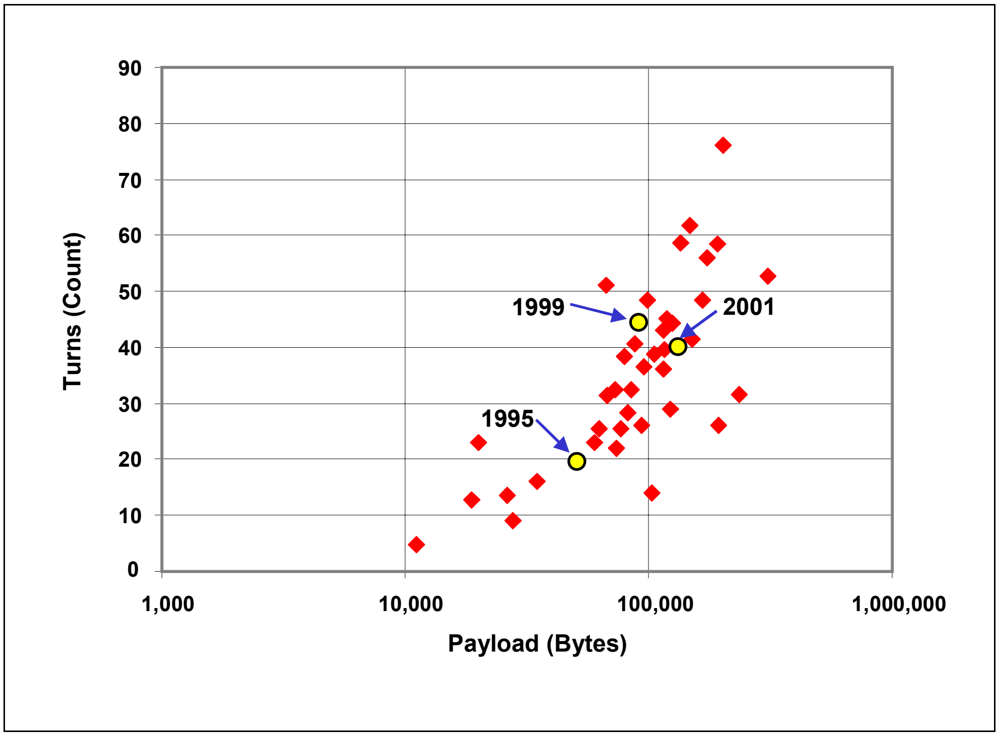

Figure 2 – Application Profiles of the KB40

We noticed that during this summer, ten sites in the KB40 were consistently ranked by Keynote as one of the top ten performing sites for that week. It is interesting to see that the average profile for those ten sites is 24 turns and 65,000 bytes of payload – half of the profile for the remaining 30 KB40 sites. Clearly a good application profile is a good step in making a site perform fast.

Figure 2 also indicates the trends in business-oriented website profiles. Payload has been climbing steadily – the compound annual growth rate from 1995 to 1999 was 13 percent, and from 1999 to 2001 it accelerated to 19 percent.

More alarming growth occurred in turns from 1995 to 1999, where the annual growth rate was 22 percent, but the turn count appears to have peaked in 2000, and it has since fallen considerably. The overall change in turns from 1999 to 2001 was a declined of 4 percent per year. It appears that web managers at these sites, who are under the Keynote microscope, have finally realized that simplifying the web page and thus reducing turn count is to their benefit.

Predicting Performance

We have developed a useful formula for predicting the performance of an application across the Internet. This formula was developed by analyzing the behavior of the protocols, as well as by extensive comparison with real data from real networks.

There are two parts of the performance equation, representing the two components of delay in retrieving data through the Internet. The *discovery component* accounts for the client/server interactions required to set-up the payload transfer. The *transfer component* accounts for the time it takes to move the payload bytes across the network.

Part 1 – Discovery (Accounting for Turns)
This term accounts for the delay incurred as the client and server set up the payload transfer, driven by the number of application turns. Since these application turns typically use very small packets, their network performance is limited by round trip delay. This portion of the total time is represented by:

$$\textbf{Discovery Time} = 2(D+C)+(D+C/2)((T-2)/M)+D\ln((T-2)/M+1)$$

The multiplexing factor M represents the ability of some applications or browsers to multi-thread or initiate more than one transfer simultaneously. While current browsers are set to a multiplexing factor of four, actual measurements show that such efficiency is rarely achieved; most browser/web page combinations operate at three threads.

The 2(D+C) at the beginning of the equation represents two round-trip delays, one for the TCP Open and one for the HTTP 'Get.' These two interactions must take place sequentially to get the base web page and discover how many elements need to be fetched. Once these two interactions are complete, the remaining components are subject to the multiplexing factor.

Part 2 – Transfer (Moving the Payload)
Payload transfer is limited either by the connection speed, or by the combination of window size and round-trip delay. Whichever is greater determines the transfer time. The equation calculates both times, and then chooses the larger value for this portion of the equation. This portion of the overall calculation is:

$$\textbf{Payload Time} = \max(8P(1+OHD)/B,\ DP/W)/(1-\mathrm{sqrt}(L))$$

The max function chooses either line delay or window delay in the numerator of the equation. Note that overhead is added to the payload. Overhead is a percentage that accounts for HTTP, TCP and layer 2 bytes that are added to the actual payload to move it through the network. If 10 percent additional bytes are required to move the payload, OHD would be set to 0.1 as is the current situation on the web.

The window size and round-trip delay affect the payload transfer, because the server is only allowed to send a window's worth of data before receiving back an acknowledgement from the client that the data was received. The acknowledgement time is limited by the round-trip delay of the connection. Although the window size is usually at least 8 Kbytes, TCP is required to send an acknowledgement after two full-size packets are received. Empirical evidence shows that setting the window size to match two full-size packets (3000 bytes) works well.

One more factor comes in to play: packet loss. Each loss of a packet causes inefficiency in the TCP interaction, which slows the transfer. The denominator of the equation models this slowdown.

Total Response Time
The total response time is the sum of the two sections above, discovery and transfer. (See "The Complete Formula.") An Excel spreadsheet with the full equation can be found on the NetForecast website, at www.netforecast.com.

The Complete Formula

R = 2(D+C)+(D+C/2)((T-2)/M)+Dln((T-2)/M+1)
+max(8P(1+OHD)/B, DP/W)/(1-sqrt(L))

$$R = 2(D+C)+(D+C/2)((T-2)/M)+D\ln((T-2)/M+1) + \max(8P(1+OHD)/B,\ DP/W)/(1-\sqrt{L})$$

Where:
B = Minimum line speed in path (bits per second)
C = Cc + Cs
Cc = Client processing time (seconds)
Cs = Server processing time (seconds)
D = Round trip delay (seconds)
L = Packet loss (fraction)
M = Multiplexing factor
OHD = Overhead (fraction)
P = Payload (bytes)
R = Response Time (seconds)
T = Application Turns (count)
W = Effective Window size (bytes)

The formula does not account for client rendering time of the payload, only the time necessary to bring in the payload itself. This formula also assumes that the server has the content being requested, and merely has to retrieve and send it. If the server is required to do an extensive database search before responding, this additional time needs to be added.

The equation is making the simplifying assumption that the server payload is much larger than the client payload, as is the case for web pages. Clearly, client payload must be accounted for in situations where client payload dominates, such as web publishing or on-line backup. A more sophisticated approach may be needed when client and server payloads are relatively equal.

The bandwidth value in these equations is the bandwidth of the slowest link in the network between server and client. This is typically the access line from the service provider to the enterprise or home.

Lastly, the mux factor must be set to match the behavior of the client or browser. As has been noted, current browsers achieve an effective mux of three. However, transaction-processing applications, file transfers and most older applications are single-threaded, and so will require that the mux be set to 1.

Measuring Web Performance

We used detailed measurement data of the top 10 performing sites in the KB40 in order to verify the accuracy of the formula shown above. This also gave us the opportunity to compare the techniques of two leading web measurement services, Keynote and Porivo.

Keynote Systems (www.Keynote.com) tests the Keynote Business-40 each day of the workweek, every 15 minutes, from each of its testing agents. These tests are then averaged together and sorted by page-load time. The top 10, those websites with the highest performance (fastest page-load times), are listed on their website each week, showing the site name and the average time to load that web page. The Keynote agents are located in 25 cities, connected to the Internet with T3 speeds or greater. The KB40 websites are tested from each server, and the performance measurement they post is an aggregate of those values.

Porivo Technologies (www.porivo.com) has thousands of clients that Internet users have downloaded into their desktop computers. Porivo is then able to schedule clients to run specific performance tests throughout the day. Porivo ran performance tests against the top 10 members of the KB40 for 4 weeks, from late July through mid-August, using agents on T1 and cable modem access lines. The data was compiled to match the Keynote measurements as closely as possible. The average download speed of each site was then calculated weekly from the results. Note that because the Porivo client is running on a user desktop, it will be affected by local proxy and caching servers. It will not, however, take advantage of browser caches.

Keynote and Porivo have different testing approaches, and can be expected to deliver slightly different results. We have compiled data here for four weeks in August from each service, and Figure 3 shows the results. In looking at that figure, however, two quite different answers emerge to the same question: How fast does this page download? Is this the same Internet?

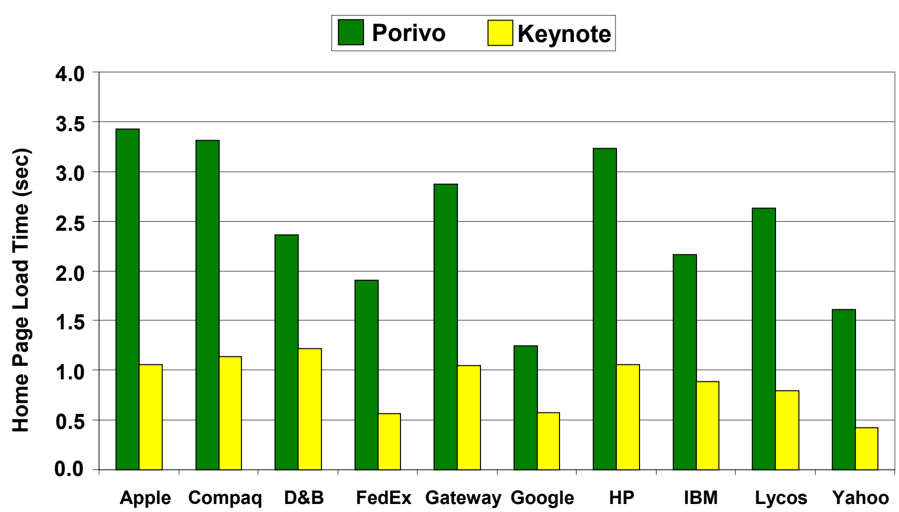

Figure 3 – Comparing Measurement Services

Lets work with the formula proposed above, to recreate these numbers. The variables we have to play with are bandwidth, client-processing time, round-trip delay, packet loss and the multiplexing factor. The web page profile and server parameters are constant across both measurement services. Table 1 shows the result of poking at the formula variables until it properly recreates the results shown in Figure 1 above. Here is why we believe these parameter changes make sense.

Table 1 - Variable Changes

	Keynote	**Porivo**
Line Rate	45 Mbps	1.5 Mbps
Loss	0.1%	5%
RTT Backbone	21 msec	21 msec
RTT Access	0	26 msec
M	6	3
Client Proc	12 msec	36 msec

First on the list is access line rate. We focused the data taken from Porivo on users with broadband access, but "broadband" often means a speed above 384 kbps, about one-forth of a T1. But Keynote has high-speed access – 45 Mbps T3 – from its data center locations. For most websites, the delay caused by payload transiting a T1 link will not be a decisive factor, but it may make a difference for low latency connections.

Because real user desktops are further from the Internet core and on slower lines, they also exhibit higher packet loss percentages. A prime spot for packet loss is at the boundary between the core and the user's access ISP. This additional loss slows the transfer, as explained above.

Thirdly, round-trip delay is lower for Keynote than for most end users. Keynote has its test agents set up at data centers around the country, where they are directly tied to backbone providers. They are never far, in Internet delay, from the big carrier that will take them to the website being tested. End users, on the other hand, are on the *other end* of an access link or even an access ISP, which causes additional delay. We have broken down the delay in Table 1 to show both the backbone and access portions to demonstrate this difference.

Multiplexing also comes into play when a testing service emulates the browser behavior. If the test agent behaves exactly like a browser, it uses only three connections at a time and, typically, downloads only two objects at a time. The Keynote agents, once they parse the base page, fetch as many objects simultaneously as possible. This tests the performance of the Internet, but does not necessarily match the user experience of opening that page.

Lastly, client processing follows much the same argument as multiplexing. Because Keynote uses a dedicated, powerful server for its testing agent, it is likely to have much shorter client-processing times than a user desktop. The desktop is running a non-real time operating system and may be doing other tasks concurrently. Increasing the client-processing value for Porivo makes sense in this context.

Making the above parameter changes and feeding the equations with the profiles of the top 10 KB40 web pages then yields numbers that closely match the empirical results as shown in Figure 4. If the formula were a perfect predictor of performance, then all the points would fall in the diagonal of the figure. Most fall within the 10 percent error band along the diagonal.

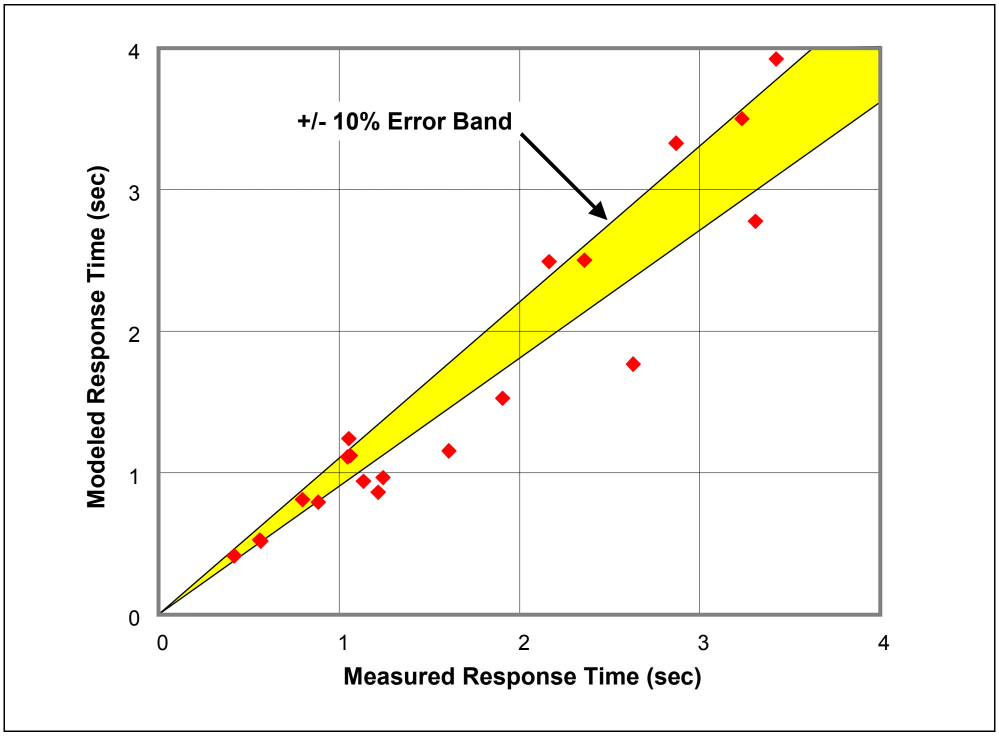

Figure 4 – Formula Verification

What Contributes to Poor Performance

Given that the formula can match measured performance of ten different sites using two different measurement techniques, we are confident of its predictive capabilities across a wider range of alternatives. It is interesting to test the value of new technologies that are proposed to improve web performance.

Here we use the overall KB40 average (115,000 byte payload and 40 turns) as a better indicator of a typical page, because the tests were clearly performed on a group of sites that have an unusually low payload and turn count. We also made a few changes to the Keynote parameter settings in order to make it a more realistic representation of a very well connected user that we call the "Best Case" as shown in Table 2. The Porivo parameters are essentially unchanged, becoming the "Typical Case." In addition, we had to create a new RTT for dial-up users that accounts for the 100 msec latency penalty of accessing the Internet over a dial-up modem.

Table 2 – Realistic Performance Parameters

	Best Case	**Typical Case**
Line Rate	1.5 Mbps	1.5 Mbps
Loss	1%	5%
RTT (Backbone + Access)	55 msec	110 msec
RTT for Modem Users	155 msec	210 msec
M	3	3
Response Time	3.9 sec	8.2 sec

The results of these changes indicate that a Best Case broadband user will see the typical web page load in four seconds, while the more Typical Case broadband user will likely experience a load in eight seconds. This range is in line with our experience.

Now that the real performance model is understood, we can vary any parameter in order to see the effect. The most logical investigation is to study the effect bandwidth has on response time. Figure 5 shows the dramatic effect to response time when bandwidth is much slower than 1.5 Mbps. The majority of Internet users see a typical page load in more than 20 seconds, a vast difference from the Best Case broadband user.

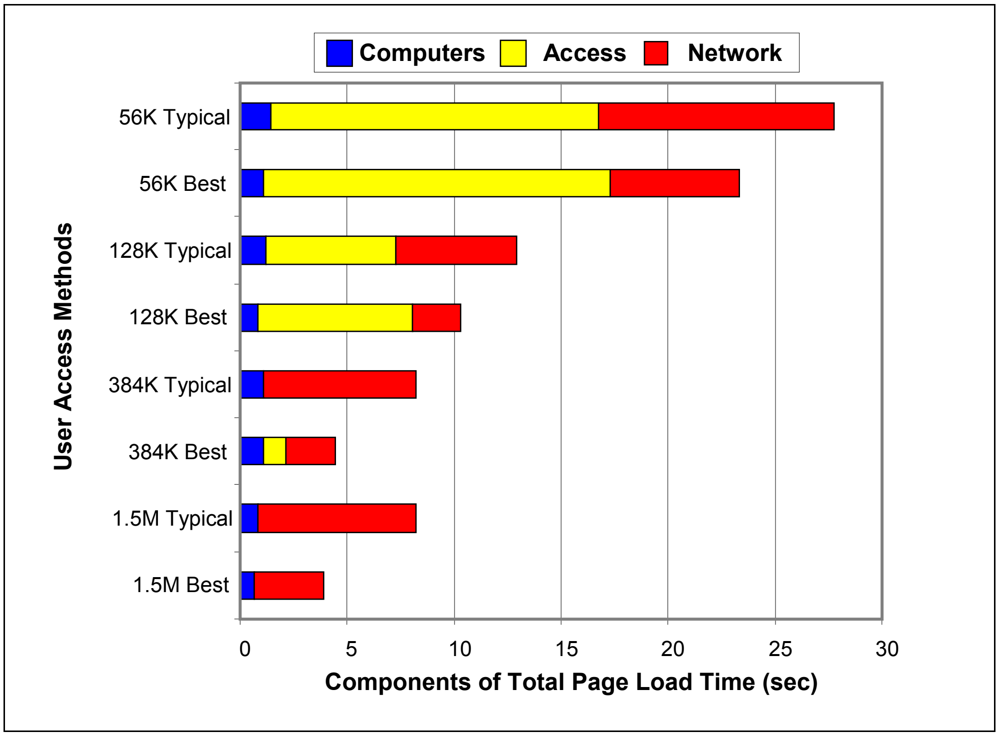

Figure 5 – Overall Delay Drivers

However, it is also interesting to make some of the elements of delay go away. Figure 5 shows the components of delay for users in each of the bandwidth classes. In each case, we recalculate the formula as if the element under investigation were perfect. We replace the access line rate (bandwidth) with a Gigabit Ethernet pipe (1,000 Mbps). In the case of the computers, we drove both the server and client computing times to zero. A perfect network is one that has no latency and no packet loss.

Each recalculation gave an equal or better result than the base case. An equal result indicates that the change to the parameter made no change to the total response time. We then apportion the improvements to the base case (Figure 5).

Note that access bandwidth improves things dramatically as you go from 56 Kbps dial-up to 384 Kbps dedicated, but then *the effect goes away completely by 1.5 Mbps*. The reason there is still some minor benefit for the 384-Kbps "Best Case" user to buy more bandwidth is that his/her network performance is good enough to take advantage of the better speed. However, there is no advantage for either the Best Case or the Typical Case user to buy more than 1.5 Mbps. In fact, the point at which no more benefit occurs is at about 512 Kbps.

As broadband access grows, the focus will have to shift to making network latency and loss commensurately lower. Packet loss can be addressed with proper engineering. However, latency is limited by the speed of light and the circuitous routing that paths will always take in a network. *The only sure way to improve performance below four seconds is to either move the server closer to the user or to reduce the number of turns in a web page.* There are many companies that are addressing these approaches with a variety of performance boosting products and services.

Since we have performed similar in-depth studies of web performance in 1995 and 1999, it is interesting to compare the components of web performance over time. Figure 6 presents the time to load the then-typical web page by the then Best-Case business user. The methodology used in the previous studies is similar the process described in this report. However, the equation used in each study was a little different reflecting improvements in our understanding of web performance.

The data in Figure 6 show that computers have been tracking Moore's Law in their ability to find and process larger amounts of payload. Access lines have been improving as broadband arrives, but the time that is a direct result of network latency and loss is becoming a significant portion of the total page load time. We can expect that the Best-Case business user will reach a performance improvement limit as the equation approaches an asymptote driven by fundamental network latency. In fact, network latency and loss already have become a critical impediment to improving performance to the speeds required for serious production applications and electronic commerce on the Internet.

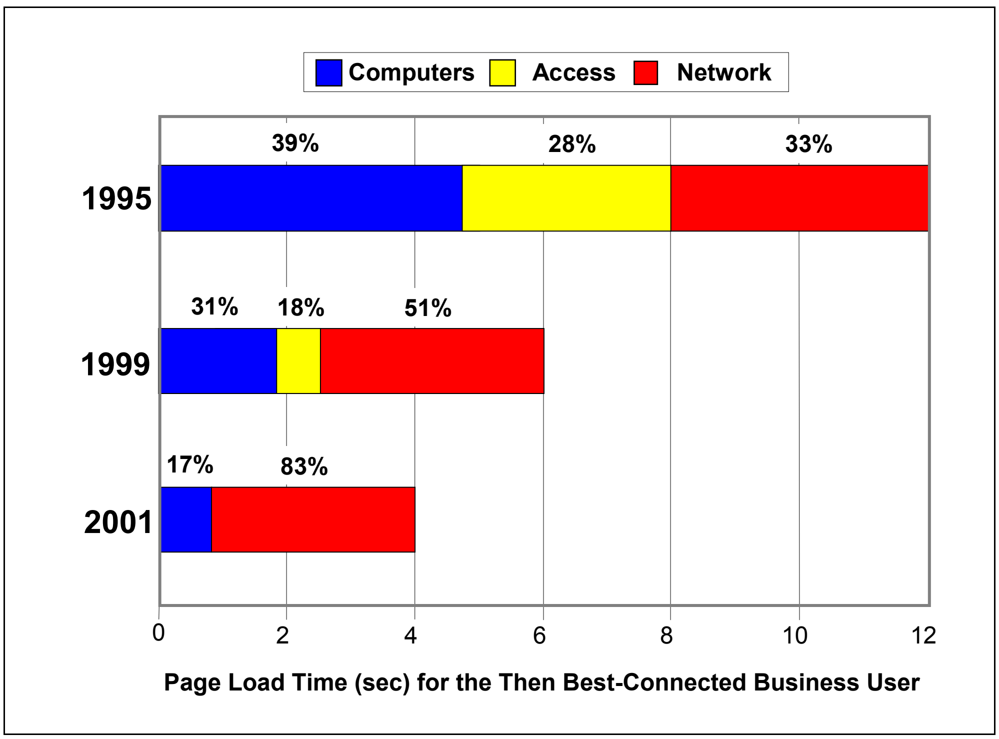

Figure 6 – Performance Factors Over Time

Implications of the Data

Clearly, there is a web performance problem. Real world performance is 3-times to 10-times slower than he often-quoted Keynote Business 40 numbers; no users actually see the performance of the KB40.

During a meeting held here at NetForecast last year (May 2, 2000) we challenged the CEO of Keynote, Umang Gupta, with the observation that the KB40 index was shifting away from being a real measure of true performance. His reply was, "Our data is not intended for use as a measure of any specific user, and they should not be used for historic trend analysis since they change over time. The data is intended as a benchmark for comparison between websites at any given time."

While Keynote is providing a very useful and interesting benchmarking service, they should remove "seconds" from the charts and simply call it an "Index" very much like the Dow Jones Industrial Index: Interesting but not relevant to daily life.

We can think of only one group of users that sees performance approaching that published by Keynote. They are connected with lightly-loaded T1 lines directly to a core ISP, have only the latest fastest desktops and spend most of their day checking on company credit ratings at Dunn and Bradstreet and chasing document shipments on FedEx; in other words, VCs.

These VCs who think the 'Net is just fine are thus making two very wrong bets: First, they over-invest in bandwidth plays (e.g., optical) while under-investing in companies that make the 'Net run better (edge services). Second, they are surprised when the mass market (millions of users) does not show up for their dot-com investments. Maybe, just maybe, the fact that the basic web page took about 20 seconds to download had something to do with it.

Porivo stands out as a reliably accurate source for realistic measurements of the true user experience, largely because of two key factors: First, its agents are on real desktops that see the performance of the full path from the server. Second, it can tap the resources of thousands of agents distributed over demographic and geographic points that match the true Internet user population.

A lot still needs to be done to improve performance on the 'Net. We also need better methods of measuring performance, along with understanding the impacts of poor performance. It will be interesting to watch the improvements emerge that successfully tackle the real long-term culprits – payload, turns count and network latency. ¤

Peter Sevcik is President of NetForecast and a leading authority on Internet traffic and performance. He has contributed to the design of more than 100 networks. Peter led the project that divided the Arpanet into multiple networks in 1984, which was the beginning of today's Internet. He can be reached at peter@netforecast.com.

John Bartlett is a Principal at NetForecast, and has 24 years of experience designing at the chip, board, system and network levels, with a focus on performance. John led the team that built the first VLAN implementation, one of the first ATM switches, and he is a leading authority on the behavior of real-time traffic on the Internet. He can be reached at john@netforecast.com.

NetForecast, Inc., helps leading service providers, enterprises, and vendors navigate the changing competitive landscape of the Internet economy. The firm specializes in developing return on investment (ROI) models to determine the business value of new Internet technologies.

www.netforecast.com