Application Response Time Improvements with Transparent Turn Reduction

By Peter Sevcik and John Bartlett
September 2003

As corporations expand geographically and use Web-based approaches to operate their businesses, acceptable application response times become more critical, yet more difficult to achieve. Almost every supplier, customer, partner, and other stakeholder interaction is performed at least partially over the Internet. Even employee interactions are moving from private networks to the Internet in order to reduce the cost of supporting these users. Networked applications increase business efficiency and widen the circle of potential business partners.

Communicating with partners across town is not like communicating with them across an ocean. Making a product catalog accessible to users in China is not the same as *booking orders* from China. To support business goals, networked applications must be available wherever a firm does business, and must also perform acceptably under less than perfect network conditions. Response time performance is key to business application success. Fast response time enables businesses to execute IT initiatives such as globalization (the ability to conduct commerce, manufacturing, development or support on a global scale) and consolidation (the ability to reduce cost and complexity by centralizing application infrastructure).

However, applications designed to operate locally or within the benign confines of a private enterprise network are now touching more end points and traveling longer distances than their developers ever imagined. Unfortunately, what many of these attempts achieve in geographic and business reach, they sacrifice in performance.

A growing cast of on-line business players, combined with increasingly stringent requirements for business applications to perform well globally, is prompting multinational businesses to address the application response time problem. This report explores how to achieve equivalent performance for local and distant users with a new approach called Transparent Turns Reduction.

Transparent Turns Reduction (TTR)

Transparent turns reduction is a new approach to improving application performance across long distances. TTR vendors recognize that application turns are the most significant factor in the performance equation once round trip delay exceeds 50 or 60 milliseconds. Offered as a service, this is how it works:

TTR solutions are bi-nodal with one node located near end users (local-node) and the other located near the origin server (server-node). Customers delegate application DNS processing to the TTR vendor. When the client computer requests the address of a Web site supported by a TTR service, the TTR vendor's DNS system responds with the address of the nearest TTR node. The initial page request then goes to this local-node, which acts as a proxy for the origin server. The local-node acknowledges the TCP session and the page request is forwarded to the TTR server-node nearest the origin server. This server-node then issues the request to the origin server, obtains the response, and forwards the first response back down the line. At this point the two TTR nodes have interjected themselves into the path between the client and the origin server with no change in the transfer process of the base page.

NetForecast Report 5066

Within this framework, there are several complementary techniques that have been used to achieve turns reduction. Some solutions address the extra round trips caused by downloading images and objects embedded within the base page. The server-node parses the base-page, predicts requests for subsequent objects, and issues the corresponding requests to the origin server immediately. All of this content is then transferred as a single transaction to the local-node. When the browser receives the base page and then requests the remaining elements of the page, they are already waiting at the local-node and are delivered as if the origin server were only a few milliseconds away.

In addition, TTR eliminates excess low-level TCP protocol turns, such as handshakes, slow-start pauses, and undue re-transmission delays, by substituting a more efficient proprietary protocol between the nodes. These protocol technologies may, under some conditions, reduce turns associated with the base-page and encrypted content.

TTR moves a part of the client functionality (browser parsing of base-page and getting all the page elements) next to the server and optimizes the delivery of the data to the local-node. The key difference between TTR and approaches that bring the content closer to the user before the user asks for it (e.g., caching, content delivery networks) is that TTR does not store content at the edge. Therefore, the content is assured to be the exact temporal form as presented at the origin. For this reason it can be applied equally well to both static and dynamic content.

This report analyzes the value of TTR along with other leading performance enhancement techniques. Such analysis must first define applications and scenarios of operation within a proper framework of application performance.

Application Response Time Performance

To accurately judge alternative performance enhancing approaches, it is essential to place them within a realistic context of acceptable, marginal, and unacceptable performance.

How Users View Performance

Our research shows that application users (regardless of whether the application is running on their own machine or over a network) judge application interaction speeds based on the following factors.[1, 2]

First - a user budgets the time he or she is willing to spend waiting for an application to respond based on past experience with similar tasks using the same application. The user's response time expectation is set when he or she completes a function and is about to wait for the response.

Second - the length of time a user is willing to wait for the application to deliver results depends on the amount of time the user expects to spend on the next activity. The longer the user expects to spend on the next step, the longer the acceptable wait time preceding that step. This means that the longer it will take to process the results delivered by the application, the more tolerant the user will be of a long response time. The time a user will take to process the results is determined by two parameters: the number of elements on the screen, and the repetition of the interactions in the session.

The results of our research show that users' perceptions of response time fall into three performance zones:

> **Satisfied** – A user considers computer interactions satisfactory when performance is fast enough not to get in the way. A simple way to define

satisfactory performance is when the user does not mention it in a critique of the interaction. In this zone, the user can concentrate fully on the task at hand.

Tolerating – A user tolerates performance with wait times that the user notices and mentions when asked. At this performance level, the user's concentration is impaired, especially in sessions requiring multiple interactions. Awareness of the time lag and diminished concentration result in lower user productivity.

Frustrated – In this final "lack of performance" zone, a user is past tolerating the response time, and changes his or her behavior towards the application. For example, a casual user may abandon the process, while the production user may stop working on an assignment. In both cases the user gives up during the period of frustrating performance. Although the user may return, the enterprise pays a price for diminished performance, or lost and/or unhappy customers.

How Web-Based Applications Deliver Performance

All computer applications operate as a series of data exchanges between the user's client or browser and the host or server running the application. The end-to-end response time across the client-network-server system is driven by the following parameters.

Two parameters determine the demands that will be placed on the network by the application – **payload and turns**. Payload is the size of the data elements sent by the client or the server. In most cases the server payload is much larger than the client payload. Application turns are the number of non-payload bearing exchanges between client and server (excluding TCP layer ACKs). Each application has a unique profile of payload and turns.[3]

Three service parameters affect network performance – **bandwidth, delay, and loss**. The effect of bandwidth diminishes as network speeds exceed 500 Kbps, leaving delay and loss as the primary contributors to response time for broadband connected users.[4]

The product of turns and delay contribute most significantly to poor response time for Web-based applications[4] in the typical business-to-business scenario where users access the Internet at broadband speeds. Since the turn count is fixed by the application, addressing the delay portion of the relationship has historically been the only way to improve poor response time. However, due to advances in routing and capacity management, Internet delays are approaching their theoretical minimums. So, new approaches, such as the Transparent Turn Reduction, are required to improve application task response-time.

How the Internet Supplies the Key Performance Parameter of Delay

Network delay and loss are a function of distance. Packets do not travel at the speed of light, nor do they travel in a straight line. Networks inject more delay and loss as physical distance increases. But delay and loss caused by the Internet is not uniform – even at a fixed physical distance. Packet networks have many alternative paths, and switch packets among multiple routers that also handle traffic of competing flows. The result is that when measured, performance is centered on an average value, but has a very skewed distribution.

Figure 1 demonstrates this phenomenon for users across the United States. This is a distribution of 100,000 measurements made from a Web site near San Francisco to all users on the East Coast of the United States. The nominal, best case, round-trip time (RTT) is about 110 msec. A few users see performance that is faster due to their network location relative to San Francisco. The curve peaks at 105 msec as expected, and has a median (half the users below or above this point) of 190 msec.

However, the overall mean (average) for the samples is 283 msec, which is significantly higher than the median. There are few users to the left of the peak, while a significant number of users fall to the right of the peak. In fact, one fifth of users experienced a delay of greater than 300 msec. The bottom line is that few users see the best case, while many users see significantly slower performance than even an average would indicate. This "long tail" effect has been verified by four different measurement services.[5]

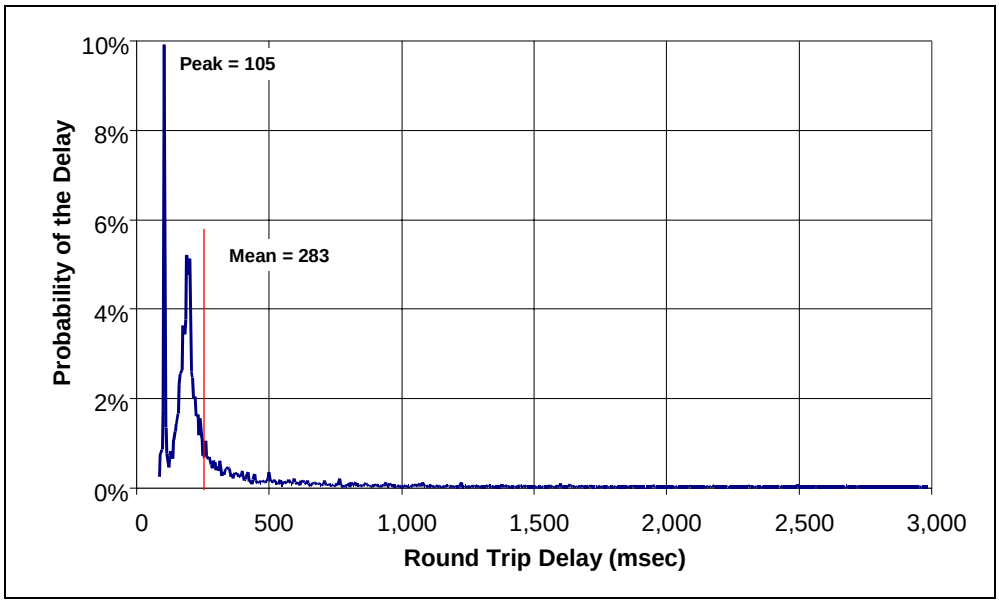

Figure 1 – Distribution of Network Delay (in 5 msec increments)

Specifying Performance of Mission Critical Applications

Performance is critical in business-to-business electronic commerce, since there is money at stake if users are not satisfied. For the purposes of this report, we focus on four business applications, each with a different satisfied response time target (see Table 1).

Customer Relationship Management (CRM) – CRM applications have a wide variety of uses, the most critical of which is in a call center. The system must be highly responsive or the operator will be forced to say the dreaded words, "I'm sorry but the system is slow today." The most frequently used tasks require the user to read about 3 elements of data on the screen, but they are called up repetitively at a very fast pace. The satisfied response time target is 3 seconds.

Supply Chain Management (SCM) – Many organizations use SCM over the Internet to expand the geographic reach of their suppliers. All suppliers should be able to access the buyer on an even performance playing field. If users in some regions of the world experience poorer performance, their suppliers' effectiveness is compromised.[6] SCM applications typically have two data elements to read, and are highly repetitive, thus the satisfied response time target is 4 seconds.

Electronic Transactions (eTrans) – These are typically financial services sites such as banking or stock trading. The user reads only one field on each page – price, order, confirmation, etc., and the repetition is relatively low for a typical

consumer, resulting in a target of 3 seconds. This pace would naturally be much faster for a trader at a Wall Street firm.

Electronic Shopping (eShop) – An example of this application is a consumer configuring and purchasing a computer on a Web site. The user reads about 2 elements per screen, and repetition is high. This yields a response time target of 4 seconds.

In Table 1, the four applications described above fall into a diagonal band of applications demanding high performance (shown in yellow). These are typical targets based upon our research and case studies. An enterprise with these applications may have different element counts and repetition rates which would yield different targets on Table 1.

Table 1 – Satisfied User Response Time Targets (in seconds)

		Number of Screen Elements User Views		
		1	2	3
Task Repetition	Low	3	6	9
	High	2	4	6
	Very High	1	2	3

The preceding definition of performance zones and application classes yields to a set of criteria for evaluating performance as shown in Table 2. Note that the tolerating-frustrated threshold is four-times the value of the satisfied-tolerating threshold which is documented in studies of the human-computer interface.[1]

Table 2 – Performance Targets

	Performance Zones		
	Satisfied (sec)	Tolerating (sec)	Frustrated (sec)
SCM	<4	4-16	>16
eTrans	<4	4-16	>16
eTrade	<3	3-12	>12
CRM	<3	3-12	>12

It is important that the applications studied in this report have profiles within the range of typical Web-based business applications. Figure 2 shows that they are in the high performance corner of Web applications yet sufficiently different from one another to show unique requirements on acceleration solutions.

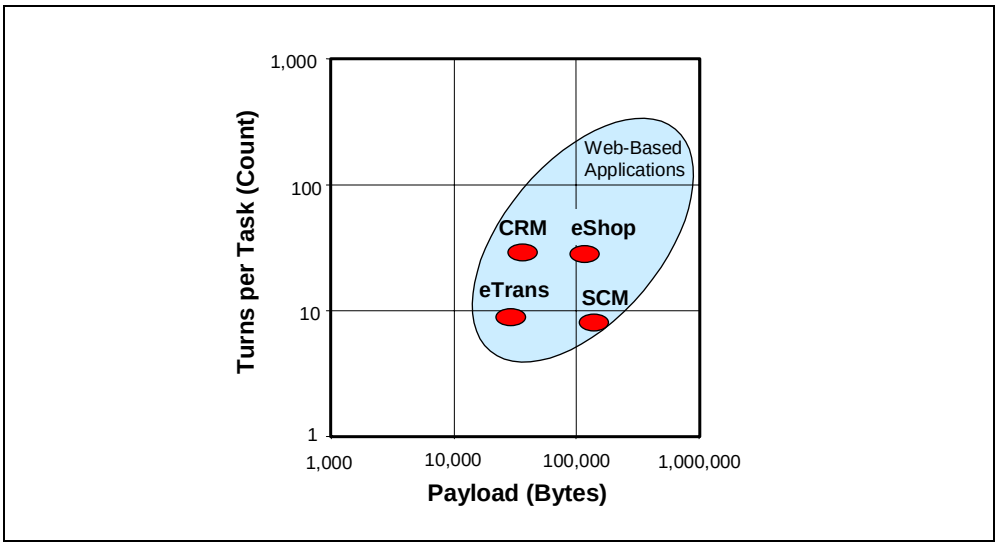

Figure 2 – Application Profiles

All of these applications provide good performance to users on a corporate campus or even on the Internet, assuming the users are relatively close to the server. However when the goal of the enterprise is to expand the number of properly engaged users to as large a community as possible, the application must provide satisfactory performance across the globe.

For purposes of this analysis, we have defined a scenario in which the servers are hosted in Los Angeles, CA. In this scenario, users in southern California experience excellent performance, and users along the West Coast to Seattle do not complain. This is a significant user population, but it does not represent the full potential of all application users, raising the question, "Can the application satisfy users across the United States and even Europe and Asia?" Answering this question requires determining performance for two critical distances: 2,500 miles and 5,000 miles, as shown in Figure 3.

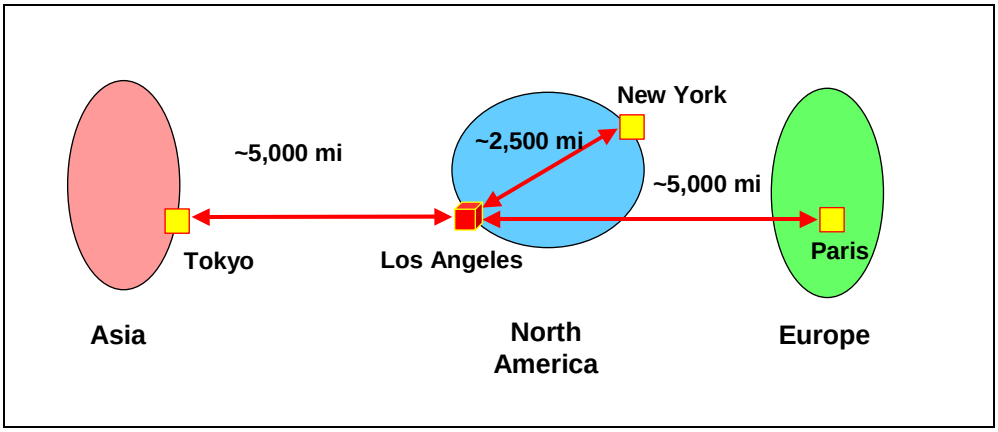

Figure 3 – The Geography Problem

Given these performance target specifications, application profiles and geographic span, we can now model the resulting response time and gauge the effects of acceleration solutions.

Modeling Alternative Acceleration Approaches

NetForecast models application behavior using a profile specific to each application. These profiles are determined by monitoring each application's traffic while an expert user executes the most common tasks. Profiles include information about the threads, the number of turns and the payload transferred by each task. Application responsiveness in different network environments can then be modeled using formulas that predict the effect of network parameters, including round-trip delay and packet loss.[4]

This study modeled three technologies: Content Delivery Networks (CDN), Compression, and Transparent Turns Reduction described above. The effect of each technology differs for each application profile, because each application is more or less sensitive to round-trip delay, packet loss, or payload size. Furthermore, each enhancing technology addresses the variables of the performance differently. Compression, for example, reduces the payload transferred by the application. CDN offerings reduce round-trip delay by delivering some of the content from a server that is local to the client.

Distance is not the sole determinant of round-trip delay. To determine the percentage of users who would have a satisfactory, tolerable, or frustrating experience with each application at the specific distances modeled, we applied a distribution curve similar to the curve shown in Figure 1 to the round-trip time parameter.

Each application was modeled for distance and the effect of the RTT probability curve, as well as the effect of each enhancement technology. The resulting response times were then compared to their target thresholds to determine how many users were in each performance zone: satisfied, tolerating, or frustrated (see Table 3).

Table 3 – Distribution of Users by Performance Zone

	Distribution of Users at 2,500 Miles					Distribution of Users at 5,000 Miles			
	SCM	eShop	eTrans	CRM		SCM	eShop	eTrans	CRM
Satisfied									
Direct	48%	0%	65%	0%		0%	0%	0%	0%
CDN	48%	82%	80%	0%		0%	55%	47%	0%
Comp	82%	65%	75%	0%		55%	0%	0%	0%
TTR	91%	97%	95%	65%		81%	92%	89%	0%
Tolerating									
Direct	45%	91%	28%	89%		85%	81%	85%	76%
CDN	45%	14%	15%	89%		85%	35%	42%	76%
Comp	14%	28%	19%	90%		36%	85%	87%	78%
TTR	7%	2%	3%	29%		14%	3%	7%	87%
Frustrated									
Direct	7%	9%	7%	11%		15%	19%	15%	24%
CDN	7%	4%	5%	11%		15%	10%	11%	24%
Comp	4%	7%	6%	10%		9%	15%	13%	22%
TTR	2%	2%	2%	6%		4%	4%	4%	13%

Table 3 shows how both the applications and acceleration techniques differ significantly from one another. The first area of interest is the "Direct" row, which represents the current approach of having users directly connect to the server over the Internet. The SCM application operates like most Web applications, with about half of the users satisfied and nearly half experiencing tolerable performance across the United States.

The remaining 7 percent of users are frustrated. The eShop and CRM applications, on the other hand, have no satisfied users across the United States. The eTrans application is performing best in the direct mode, with 65 percent of its users satisfied.

The picture is much worse when overseas users access these applications where, none of the applications has satisfied users. Most users experience tolerable performance, while 15 to 25 percent of the users are frustrated. All of these data show that in order to be successful across these distances, some form of acceleration is essential.

CDNs help eShop and eTrans applications because they contain a lot of static content (photos, graphics) associated with each task. The ability of the CDN to pre-deploy static content and deliver it with a shorter delay path is evident. However, the CDN does little to benefit the other applications.

Compression is beneficial for big payloads that cannot be cached, but can be compressed as is the case with SCM. Again, the benefit is restricted to this application and does little for the other applications.

Transparent Turns Reduction shows the greatest benefit for all the applications, and it shifts the highest number of users into the satisfied performance zone. It is surprising how much of an effect it has at the shorter 2,500 mile distance where companies are operating B-to-B applications across the United States.

Another way to examine these results is to track how many users were moved from a poor performance zone to a better performance zone. Figure 4 shows the relative shift of users from frustrated to tolerating or satisfied, plus the shift from tolerating to satisfied for each acceleration technology. Again, TTR provides the greatest benefit to the largest number of applications and distances.

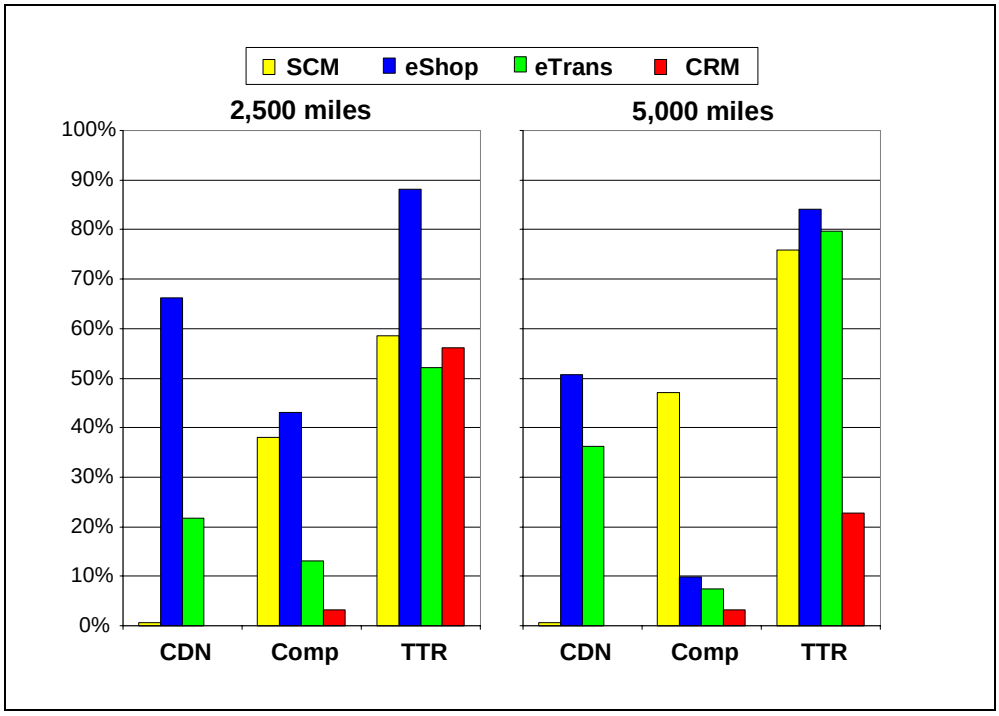

Figure 4 – Relative Shift of Users to a Better Performance Zone

Summary

Enterprises should set critical performance targets and study the distribution of users within those targets. If done realistically, there will likely be a set of users in the frustrated zone. The critical performance targets provide a foundation for defining a performance policy, reporting on performance, and evaluating acceleration techniques.

We have been writing about the hidden performance demon of turns since 1996. Despite our calls to reduce turn counts of typical tasks, they keep climbing. The good news is that finally there are practical solutions to the problem through the use of TTR. What distinguishes this approach from previous attempts to deal with turns is that TTR is transparent. Neither the server nor the client needs to change anything.

Enterprises will do well to look into TTR as they expand their Web-based applications to reach more users over greater distances. This will help ensure that the business push to serve new distant markets succeeds. Of course, a TTR product or service may include some additional technologies such as compression in order to provide a more comprehensive solution to the performance problem. Enterprises should carefully evaluate the acceleration features mix appropriate to the applications and user scenarios they are supporting. ¤

References

1. Sevcik, “Understanding How Users View Application Performance,” *Business Communications Review*, July 2002
2. Sevcik, “How Fast is Fast Enough,” *Business Communications Review*, November 2002.
3. Sevcik, "Designing a High Performance Website," *Business Communications Review*, March 1996.
4. Sevcik and Bartlett, “Understanding Web Performance,” NetForecast Report 5055, October 2001.
5. Sevcik, “Web Performance, Not a Simple Number,” *Business Communications Review*, January 2003.
6. Sevcik, “Accelerating e-Commerce,” *Business Communications Review*, November 2002.

Peter Sevcik is President of NetForecast and a leading authority on network technology. He specializes in performance analysis and holds the patent on application response time prediction. He can be reached at peter@netforecast.com.

John Bartlett is Vice President of NetForecast, and has 25 years of experience designing at the chip, board, system, and network levels, with a focus on performance. He can be reached at john@netforecast.com.