Service Level Agreements for Business-Critical Applications

By Peter Sevcik
January 2008

Despite lip service paid to business-IT alignment, most companies struggle to make it happen. One reason for the struggle is a single-minded focus on application development, with operations often treated as an afterthought. Of course a new application must work, and it must be delivered on time and within budget—but it must also meet the performance needs of users over time. IT and application managers often assume that if an application performed adequately during its pilot, without further ado it will continue to perform adequately later, even much later. This is a dangerous assumption.

Smart businesses do not rely on assumptions—they rely on sound information, good communications, vetted processes and meaningful quality standards. In the case of application performance, smart businesses are starting to rely on application performance service level agreements (SLAs).

Traditional SLAs focus on narrow aspects of infrastructure rather than the comprehensive application experience. Because businesses run on applications, new SLAs must be defined at the application level to properly support the business. Application performance SLAs support business goals such as increased revenue, higher productivity and more efficient resource allocation. But understanding and reaching consensus on an SLA for application performance is non-trivial. This report describes a simple process for achieving that consensus.

Because applications drive the enterprise, a variety of motivations may propel you toward application SLAs. Whatever the motivation, it is important to remember that SLAs have bottom line results. It is also important to recognize that SLAs are an emerging tool for measuring and optimizing business critical applications which will become even more critical as new application initiatives are launched. Some businesses may choose to take small initial steps and simply track performance against targets, while others may push all the way to formal SLAs. This is a journey towards better application performance management, and even small steps deliver immediate value. The most important step is the first one—so we suggest that you take that first step by reading on.

Application SLA Process – The Big Picture

An SLA process encourages your enterprise to view performance into the context of your unique business drivers. A successful process requires you to:

- Identify your key business applications (e.g. voice, file access, ERP, CRM, video conferencing).
- Define response time targets and SLAs.
- Measure performance using application performance monitoring tools that identify the different applications and business processes.
- Analyze results.

Accelerating select applications 10 times or even 100 times some of the time may lead you to think you are experiencing great results—but these results may actually be unnecessary or even counterproductive when evaluated within the context of a business-centric SLA. Furthermore, voice and video applications cannot be judged by a simple "10X" or "100X" improvement metric. Quality perceived by the user is much more complex than a mere ratio can convey.

NetForecast Report 5091

Enterprises using internal SLAs often discover that to meet their goals some applications need to be accelerated over the WAN using a distributed application delivery system (DADS), also known as WAN optimization or acceleration [1]. These solutions directly affect application performance SLA results described in this report. A comprehensive DADS solution should include control and acceleration—with the ability not only to measure, but also to provide enough granularity to isolate specific applications and processes. Enterprises should take these effects and requirements into account when evaluating DADS solutions.

Unfortunately scant attention is paid to application SLAs. Many companies move to implement an SLA when a major IT change is in the works such as rolling out a new business application, outsourcing a part of the delivery system, consolidating data centers, or converging voice and video onto the data network. These changes often hurt application performance, and an SLA reporting system can help ensure good performance.

Enterprises should not wait for a "big change" to implement an SLA program. First, it takes time to define an SLA and put an application under SLA management. Enterprises often think to add an SLA at the last minute when worry sets in that the big change may adversely affect performance. Unfortunately there is simply not enough time to get an SLA in place.

Second, without SLAs before the system change there is little if any performance baseline data. Performance may degrade in many ways when the big change occurs. You need about a year's worth of historic performance data—in SLA terms—to accurately assess performance effects. Furthermore, other applications may unexpectedly be affected by the big change. Again, the lack of comprehensive baseline data makes this difficult to detect.

Enterprises need SLA systems now. Although such systems help manage future change, you should not wait until change is imminent to begin. Change is certain, so the prudent way to manage performance given change is to deploy an SLA program today.

An SLA program enables you to manage all business applications under a standardized system. Such a system must be simple for everyone—particularly senior management—to understand. Potentially confusing technical measurements must be normalized into easy to understand metrics, and the system must provide uniform performance for key applications based on business needs. There should, for instance, be no second-class users based on geographic location.

Application Performance SLA Requirements

A service level agreement is a contract negotiated between a service provider and customer, and a successful SLA program requires a strong foundation.

A strong foundation requires SLA objectives built using industry standards. One such standard, the Application Performance Index (Apdex), is designed specifically to support business-centric application performance SLAs. An open standard developed by an alliance of companies, Apdex defines a method to report, benchmark and track application performance. This report uses Apdex methodology to define response time SLAs.

A good application performance SLA has the following properties or agreement **terms**.

- A formal application performance **threshold** or objective. Service at or better than the threshold meets business objectives. The Apdex standard defines this threshold as the *target response time*, and refers to it as "*T*"**.**
- A **score** that is an aggregate measurement that describes how well the system performed relative to the objective. The Apdex standard describes this as the *Apdex score* (or results of the Apdex formula).

- A service provider must then meet a service **goal**. Since Apdex is a numeric value between 0 and 1 where 0 is complete failure and 1 is perfect performance, the goal is defined as service above an agreed upon Apdex score (e.g., the service must stay above Apdex 0.90 [T]).
- It is unrealistic to expect service to achieve the goal under all circumstances all the time, so an SLA also describes **conditions** when the SLA is not in effect. The conditions usually define periods reserved for system maintenance or other periods during which the goals are put on hold.
- Finally, the SLA must describe **consequences** for falling below the goal. Without consequences, there is little incentive for the service provider to adhere to the SLA. Such consequences can be financial penalties for not meeting or financial bonuses for exceeding the goal. Within an organization penalties and bonuses are often linked to the remuneration of employees responsible for service delivery.

An SLA must be a written document agreed to by all parties involved. Enterprises should ensure that the properties defined above are included in the agreement, with other terms and conditions added as appropriate.

How Apdex Works

Apdex reports are arrived at using a four-step method.

Step 1 – Select Target Time

Define a target response time of T seconds as described later in this report.

Step 2 – Measure Performance

Measure performance and place each measurement into one of three user-perceived performance zones: satisfied (Sat), tolerating (Tol), or frustrated. Frustrated times are above F seconds, where F=4T. Tolerating times are between T and F.

Step 3 – Apply the Apdex Formula

Sum the incidents of response times that fall into each zone, and divide by the total number of measurements using the formula:

$$Apdex = \frac{Sat + Tol/2}{Total}$$

Step 4 – Show Results

Show the Apdex Index value (on a scale of 0 to 1) together with the corresponding threshold T.

See www.apdex.org to learn more.

Defining Response Time Targets

A response time based SLA measures how long something takes, so you can't apply an SLA without first measuring performance. The nature of an SLA is driven by the measurements used to enforce the SLA, so a good measurement and reporting tool is essential.

The first question when defining a user response time SLA is, what is being measured as "response time"? An application running over a network is complex, and it is important to understand the intricacies of application/user interactions to determine what to measure. Apdex defines the terms below to deconstruct the many client/server interactions and protocols to deliver information to the user [2].

Table 1 – Taxonomy of Transactional Computer Applications	
Apdex Term	**Description**
Task	Each user interaction with the application during a Process. Task time is measured from the moment the user enters an application query, command, function, etc., that requires a server response to the moment the user receives the response such that they can proceed with the application. Often called the "user wait time" or "application response time".
Turn	Each application client-server software interaction needed to generate a user response or Task (see above). These software-level client-server interactions add to the time it takes for the software to complete a Task. The user does not see Turns operating. A Turn is a client-server request-driven round-trip. Often called application "chattiness".

The degree of interaction responsiveness as experienced by a human user is paramount in application performance SLAs. Humans are able to perceive the responsiveness of a *task*, and *task time* is the time that a user could measure with a stopwatch when interacting with the application. Because it is the task time that matters to the user, performance target times should be defined using task times.

More tools measure turns than tasks, but fortunately that is not an issue because you can generate a turn-level equivalent to a task-based target time. This report describes how to arrive at a target time using either approach.

Task-Level Target Time

For starters you need to define target times at the task level, and this requires a structured approach. Target times depend upon the nature of the user's interactions with the application, which means you must understand how users perceive the responsiveness of the application given their activity.

Human/machine interface research suggests that users place task performance into one of three groups [3], each associated with a distinct set of perceptions. Each performance group is called a *performance zone*. A performance zone is defined by an upper and a lower threshold. The Apdex standard defines three performance zones:

> **Satisfied Zone:** Response times are fast enough to satisfy the user who is therefore able to concentrate fully on the work at hand with minimal negative impact on his/her thought process. This represents the time value (T seconds) below which users are not impeded by application response time.
>
> **Tolerating Zone:** Response times exceed the threshold at which the user notices how long it takes to interact with the system and potentially impair the user's productivity. Within the tolerating zone response times are greater than T, causing the user to notice that performance is lagging but to continue the process.
>
> **Frustrated Zone:** Response times reach the threshold at which the user becomes unhappy with slow performance. In the frustrated zone the user is likely to abandon a course of action or cancel a task.

To obtain a performance score, the Apdex standard applies a formula incorporating the counts for instances of all three performance zones using a sliding scale of credit for achieving the target. Each result within the satisfied zone receives full credit, while results in the tolerating zone receive half credit, and results in the frustrated zone receive no credit. See "How Apdex Works" sidebar. To illustrate how this approach works, Table 2 shows typical T values for NetForecast client engagements.

Table 2 – Satisfied Zone Target Task Times	
User Activity	**T [task] (sec)**
Business-to-Business (B-to-B)	6
Business-to-Consumer (B-to-C)	10
Customer Relationship Management (CRM)	3
eMail	9
Enterprise Resource Planning (ERP)	2
File Server Access	12
Supply Chain Management (SCM)	4

Note that these user activities may be supported by a variety of applications. Note too that it is ill advised to use an application class or specific brand to set T because each application must be assessed based on its own characteristics and uses.

Turn-Level Target Time

Some enterprises are better equipped to measure turns rather than tasks. This is often the case when gathering data from within the network. However, the same principles apply. Task target times can be converted to turn target times with the following process.

To do this, it is important to understand the nature of the specific application and how it operates. Each application has a unique profile that represents what it does over a network to deliver a task. The profiles vary greatly as Figure 1shows.

Both axes in Figure 1 use log scales, meaning that each major division constitutes a tenfold increase. The profiles are, therefore, vastly different from one another. Fortunately, however, profiles for individual applications remain relatively constant. Even software revisions generally do not change the basic profile since revisions rarely change the fundamental application architecture.

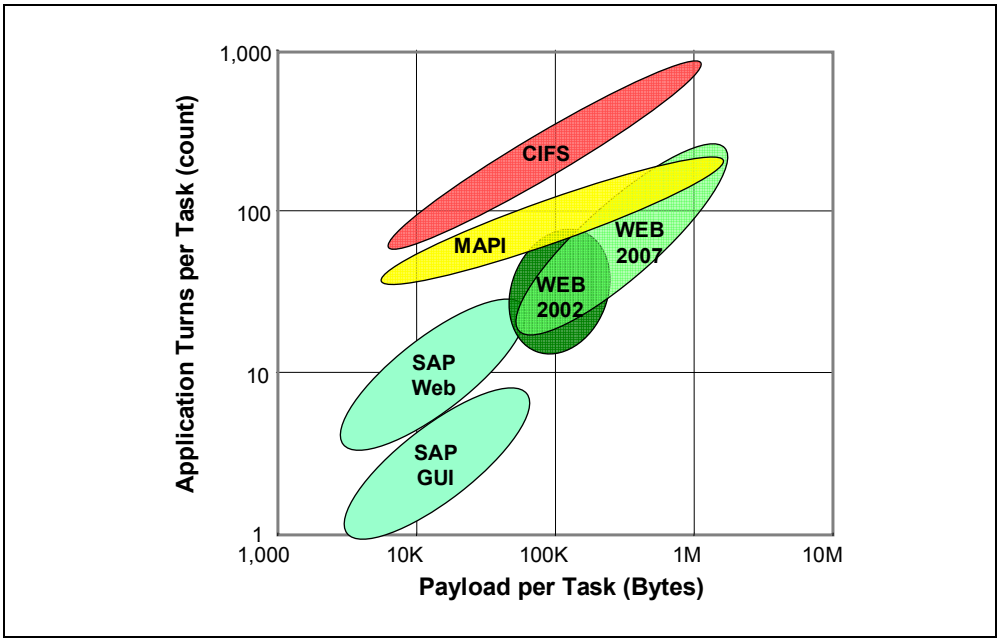

Figure 1 – Application Profiles for a Typical Task

Some applications in Figure 1 do shift due to architectural change. For example, the traditional SAP graphical user interface (GUI) implemented in the older proprietary client is very efficient—in some cases requiring only one turn per task. When SAP was re-architected to take advantage of the browser, however, the number of turns increased. This is a typical result from "webifying" applications.

Another change in profiles over time is caused by evolving use of the Web. In 1995 when simple static Web pages were the rule, the typical page required 50,000 bytes and 20 turns. NetForecast has profiled typical business Web sites over time, and found that turns per task have experienced a 13 percent compound annual growth rate—with no end in sight.

Figure 1 includes two interesting snapshots. In 2002, business applications were characterized by the browser talking to a single Web server that then communicated with a second tier of application servers. The pages themselves often required a large complex base page that set up the interaction followed by a number of items that were often unique to the specific user inquiry.

The typical Web page of 2007, on the other hand, is loaded by a dynamic client communicating with many severs to gather content [4]. Today a median of 6 servers supply parts of the typical public-facing business page. Furthermore, the dynamic client assembles a rich media product that requires significantly more content to complete the page. This shift explains the different Web profiles for Web 2002 and Web 2007 in Figure 1.

Once you your application's profile, you can convert task-based to turn-based application target times using the following formula:

$$T[turn] = \left(\frac{T}{Turns} \right) x(1 + M) x1000$$

Where:

T[turn] is the target time for satisfactory user response time (milliseconds)

T is the task target time for satisfactory user response time (seconds)

Turns is the typical number of turns per task for the application

M is the multiplexing factor (multithreading) associated with the application

If your application matches those in Figure 1, you can use the figure to derive the approximate number of turns, otherwise you must profile the application to determine the turn count. You will also have to estimate the effectiveness of multithreading in the way your clients and servers are configured. NetForecast has found that M can range form 0 to 3. Three is the maximum value of the typical Web browser (despite frequent claims that they can operate 4 threads at a time). A safe bet is that M is between 0 and 1. A T that is based on turns—T[turn]—is a cross product of Table 2 and Figure 1. Table 3 shows representative ranges of the formula results.

The turns-based values in Table 3 can be used in the same Apdex formula as that defined for tasks in Table 2. Meaningful results can be achieved but more care is required and measurements must be more precise.

Turns-based Apdex calculations are an approximation of the task-based result. The approach described here assumes that the turn time distribution is somewhat uniform. For example if a 50-turn task has one consistently long turn (e.g., large base page, flash animation), then the turns-based result will show a 2% poorer score relative to the task version. A greater number of long turns relative to a smaller pool of short turns will make

the error commensurately larger up to the worst case of a bi-modal distribution of turns. Again, it pays to know your application's profile.

Table 3 – Satisfied Zone T [Turn] Example Ranges		
	T [turn] (msec)	
User Activity	**Low**	**High**
Business-to-Business (B-B)	150	800
Business-to-Consumer (B-C)	100	300
Customer Relationship Management (CRM)	160	600
eMail	60	200
Enterprise Resource Planning (ERP)	100	500
File Server Access	20	100
Supply Chain Management (SCM)	100	400

Turns-based performance management is like trying to maintain a 60 mile per hour vehicle speed (task time) using only your tachometer (measuring turns). It can be done but it requires experience and care.

Getting to a Service Level Agreement

When you report performance scores based on an agreed upon threshold, you cover two of the five terms required for a comprehensive service level agreement. Meeting the next three terms—a service goal, SLA conditions, and consequences for meeting or not meeting the SLA—requires participation from more players.

Ultimately the application performance SLA will be a document agreed to by IT and management of the business unit the application serves. As you work your way across your business-critical applications, you will probably need to negotiate application performance SLAs separately with different business managers.

Negotiating SLAs is best undertaken as a formal process endorsed by "C" level management (e.g., the CIO). At this point, your methodology, reasoning and conclusions will likely be scrutinized. In particular, the target T will likely be questioned.

It is critical to achieve agreement on the service goal. Such agreement often takes the form of a statement like, "we will ensure that the Apdex score for this application will be above 0.90 [T] for all company locations." Once agreement on the service goal is reached, the remaining SLA terms (conditions and consequences) can be tackled. Corporate management is likely already familiar with these remaining terms from other policies or procedures.

SLA for Voice over IP (VoIP)

Telephone calls are migrating from public telecommunications carriers to corporate IP networks. The methodology described in this report can easily be applied to VoIP including an Apdex report. Voice has a long-standing standard to quantify the quality of the user experience called the mean opinion score (MOS) that was developed by Ball Labs.

MOS provides a numerical indication of the perceived quality of received media after compression and/or transmission. The MOS is expressed as a single number in the range 1 to 5, where 1 is lowest perceived quality, and 5 is the highest perceived quality.

The Telecommunications Industry Association Technical Services Bulletin 116 "Voice Quality Recommendations for IP Telephony" defines the thresholds for "tolerating" and "frustrated" voice quality at MOS values of 4.0 and 3.1 respectively.

MOS measurements can be derived from packet-level loss, latency and jitter measurements of the voice bearing UDP streams. Several traffic management and measurement vendors calculate an effective MOS result based on these underlying statistics.

SLA terms should be relevant to business needs, and should not be based on abstract notions. Cost savings, increased revenue, and better productivity are among fundamental business metrics that can link application performance to business needs such as:

- Faster checkout times resulting in improved repeat business
- Fewer help desk calls resulting in savings due to fewer help desk employees
- Faster inventory processing and order fulfillment leading to increased revenue
- Faster data input resulting in improved employee productivity
- Faster conversion of users from old to new IT systems resulting in cost savings and increased employee productivity
- Fewer uses of costly alternate communications (like shipping tapes)
- Faster back-office processing resulting in improved process efficiency

Once an approach is proven for several applications, management can address the range of business-critical applications as a portfolio. In so doing, an SLA for one application may be revised in the context SLAs for other applications in the portfolio, and higher standards can be applied to the most business critical applications. Managing SLAs for an application portfolio is achievable once fundamental standards are in place. Then and only then is application performance and business alignment truly achieved.

Role for WAN Optimization in Application Performance SLAs

Satisfying application performance SLAs for critical applications generally requires a WAN optimization solution. Because WAN optimization distorts the performance data as measured in the data center or on the WAN, true performance must be measured in the remote office closer to the user. Since WAN optimization devices are often located in the remote office, they can be a good source of the data required to measure service levels. These devices can also provide before and after statistics showing the impact that they have on the traffic.

Your SLA program should include evaluations of WAN optimization as a possible means to meet your SLAs. While you are designing your application performance SLA program, you should evaluate the measurement capabilities as well as the optimization capabilities of WAN optimization solutions.

References

1 – "Field Guide to Application Delivery Systems" by Peter Sevcik and Rebecca Wetzel, NetForecast Report 5085, September 2006

2 – "Application Performance Index (Apdex) Technical Specification," Version 1.1, Published by the Apdex Alliance, Inc., January 2007

3 – "Understanding How Users View Application Performance" by Peter Sevcik, *Business Communications Review*, July 2002

4 – "Dynamic Clients: The Network Palette Arrives" by Peter Sevcik, *Business Communications Review*, July 2007

Peter Sevcik is President of NetForecast and is a leading authority on Internet traffic, performance, and technology. Peter has contributed to the design of more than 100 networks, including the Internet, and holds the patent on application response-time prediction. He can be reached at peter@netforecast.com.

NetForecast helps enterprises understand and improve the performance of networked data, voice, and video applications.

Additional information is available at: www.netforecast.com