Best Practices for
SaaS Adoption in the Enterprise

White Paper

ThousandEyes👁
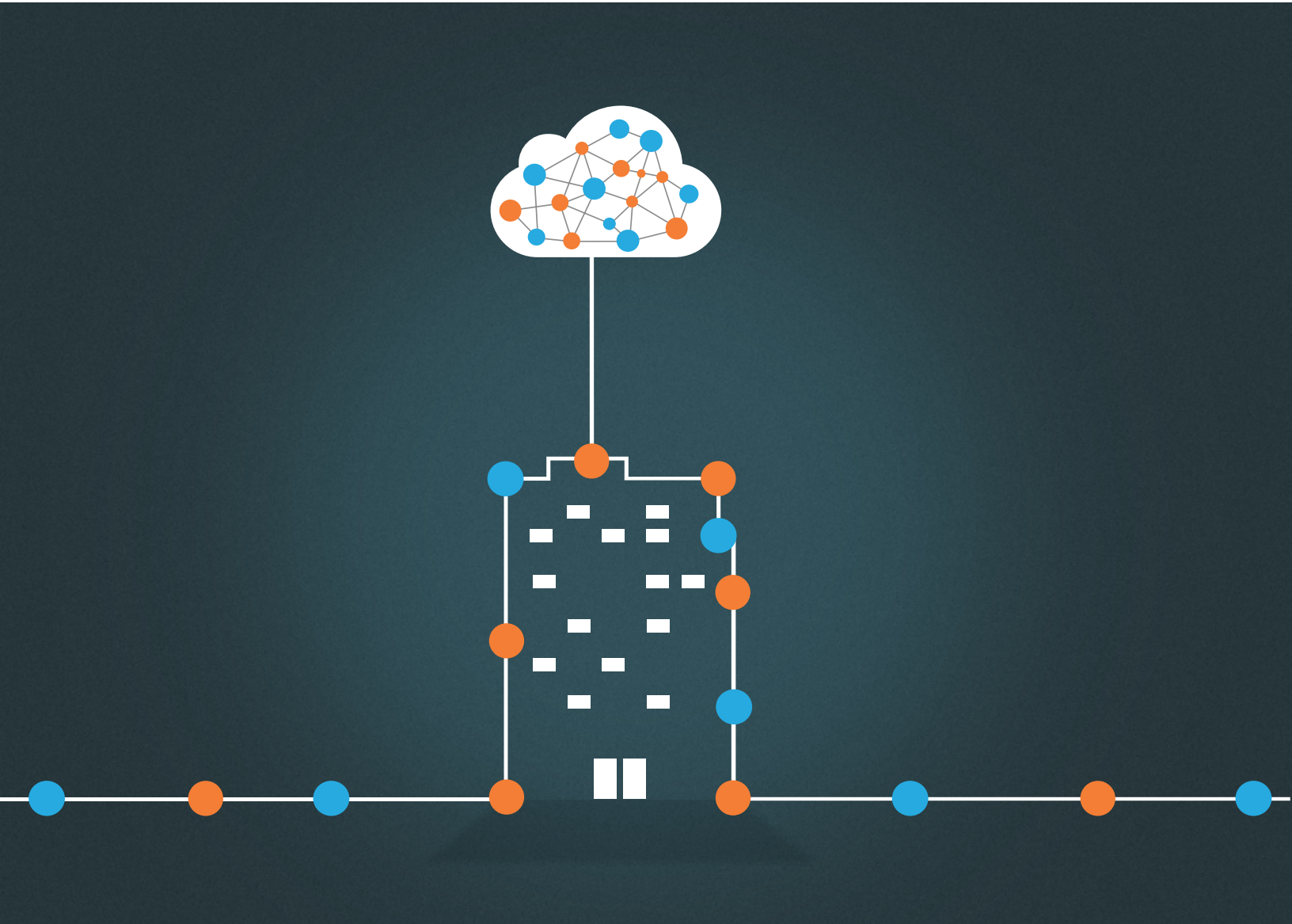
Summary

Traditional performance management solutions were built for applications owned by the enterprise and run inside the corporate network. The explosive adoption of Software-as-a-Service (SaaS) and cloud applications have brought new performance management challenges that push the limits of the enterprise infrastructure and break these traditional approaches. In this white paper, we examine why existing solutions don't work for the modern enterprise and discuss a fundamentally different approach that is needed to address the new challenges.

Moving from On-Premises Software to SaaS

Enterprises are undergoing a major transition. Software-as-a-service (SaaS) applications such as Salesforce, Workday, ServiceNow and Box are replacing traditional on-premises solutions. Gartner finds that SaaS adoption rates vary from <5% to >50%, depending on the application type.[1] Enterprises are indeed overcoming their reluctance to ship data out of their environment and adopt innovative software, platform and infrastructure services that are hosted outside their enterprise walls.

SaaS deployment models reduce the CAPEX investment associated with on-premises software solutions and data center infrastructure, and shift the costs to a pay-as-you-go model. This shift, however, relies on connectivity from the corporate network to the Internet. Applications that once ran inside a controlled corporate network are now running in any number of locations in the cloud, reliant on the best-effort nature of the Internet.

The Challenge with Traditional Enterprise Architectures

Most enterprise networks were designed for branch offices to access applications—often internally developed and deployed—hosted inside the corporate environment. According to Gartner, "WAN design practices of the past 10 to 15 years have focused on consolidation and centralization of applications and security controls in the corporate data center." For various reasons, security and control being the foremost, traffic to and from the public Internet is channeled through a small set of ingress and egress points. This Internet traffic backhauling can degrade the user experience of SaaS

"ThousandEyes has become one of the key tools in our operations center for troubleshooting network and application performance problems."

Ivan Batanov
VP Engineering
ServiceNow

1   Gartner. Hype Cycle for Software as a Service, 2015. July 29th, 2015.
2   Gartner. Optimize Enterprise Networks to Improve SaaS Performance. April 25th, 2013.
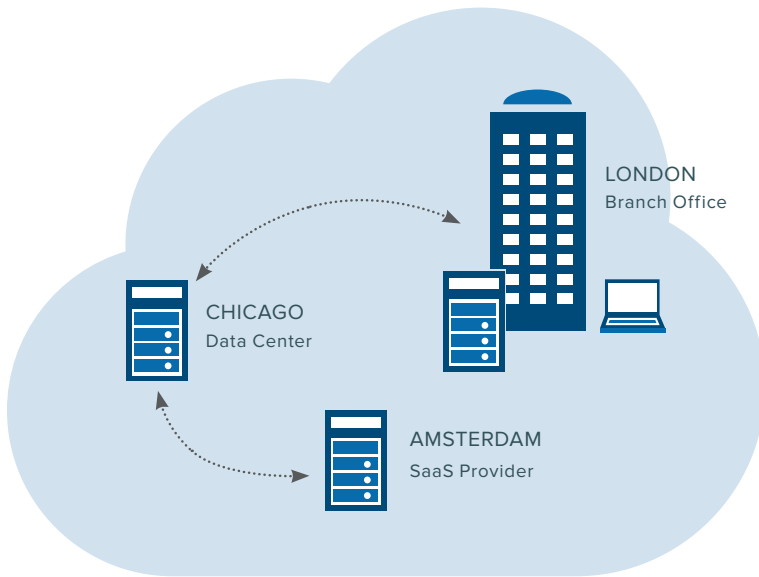
Figure 1: Backhauling of Internet traffic creates performance challenges for SaaS applications, including increase of round-trip times.

applications, often causing hard to debug performance issues. For example, in Figure 1, someone in the corporate HQ in Chicago contracted an HR productivity tool to be used worldwide. This means a user from the London office may actually be routed to a designated data center in Chicago even though the SaaS vendor has points of presence in Amsterdam. This could equate to unnecessary cross-Pacific round trips for every packet delivered.

The situation becomes even worse for chatty applications. The impact of the round-trip time (RTT) is exacerbated in the case of chatty protocols that require N rounds to complete a transaction. A transaction that would take Nx5ms inside a LAN could take Nx100ms over the public Internet, making the application virtually unusable. Available bandwidth also becomes a problem when backhauling Internet traffic, since more traffic needs to go through limited access capacities inside the enterprise (where price per Mbps is much more expensive). On the other hand, Internet transit is cheap, but the quality of service is on a best-effort basis. An interesting example is SaaS-based email; a large file sent over email can get replicated X times over the Internet link while it's being received (Figure 2).
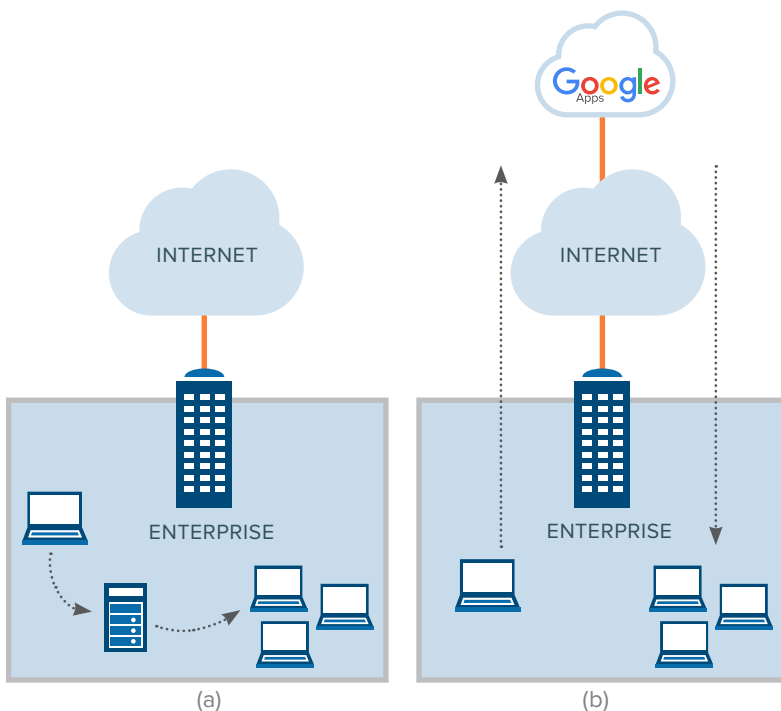
Figure 2: (a) Traditional on-premises email solution: attachments are downloaded by clients using the enterprise network. (b) SaaS-based email: attachment goes through the Internet connection N times, one for each client.

> Enterprises should evaluate application performance as part of the vendor evaluation and selection process. This is especially critical when users will be in a variety of branch offices throughout the enterprise.

Network Performance of SaaS Applications

The end-to-end performance of a SaaS application is highly dependent on three main infrastructure segments: the corporate network, the Internet and the SaaS provider. The end-to-end performance of a SaaS application and the perceived user experience depend on what happens in each of these segments.

The key network metrics are:

Latency: Defined as the round-trip time (RTT) between the client and server, network latency depends mostly on the physical distance between the endpoints, as well as network congestion. Higher latencies cause higher application response times, and lower TCP (Transport Control Protocol) throughputs. TCP throughput is inversely proportional to the round trip time (WindowSize/RTT). Also, because of the slow start mechanism, TCP can be quite inefficient in taking advantage of available bandwidth.

Packet Loss: Packet loss can trigger TCP retransmissions and lower the throughput of the connection. The throughput of a TCP connection is roughly proportional to $sqrt(p)$ where p is the packet loss probability. So a connection with 2% loss has about 70% of the throughput of a connection with 1% loss. In fact, it can be shown that in steady state the window size of a TCP connection with 1% loss is fewer than 10 packets. If each packet is 1,500 bytes long and the RTT is 100ms, the throughput is implied to be 15 KB/s.

Capacity and Available Bandwidth: *Capacity* between two endpoints is the maximum data rate that can be achieved in the absence of any cross traffic. Cross traffic will use a portion of the capacity, and the remaining amount is *Available Bandwidth*, which determines the speed of TCP connections. In general, available bandwidth in enterprise networks is limited more by cost and less by technology. Currently, WAN MPLS access can cost anywhere from $300/Mbps/month to $1,000/Mbps/month. (Note: Access to broadband Internet is 30x less expensive in comparison.) For reasons including security and control, most enterprises still backhaul Internet traffic through their data centers using MPLS, which consumes expensive additional bandwidth in the corporate backbone.
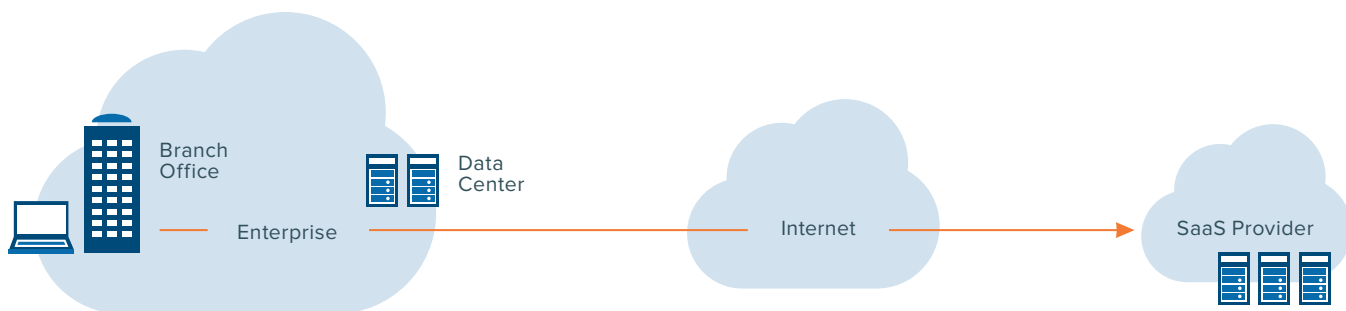

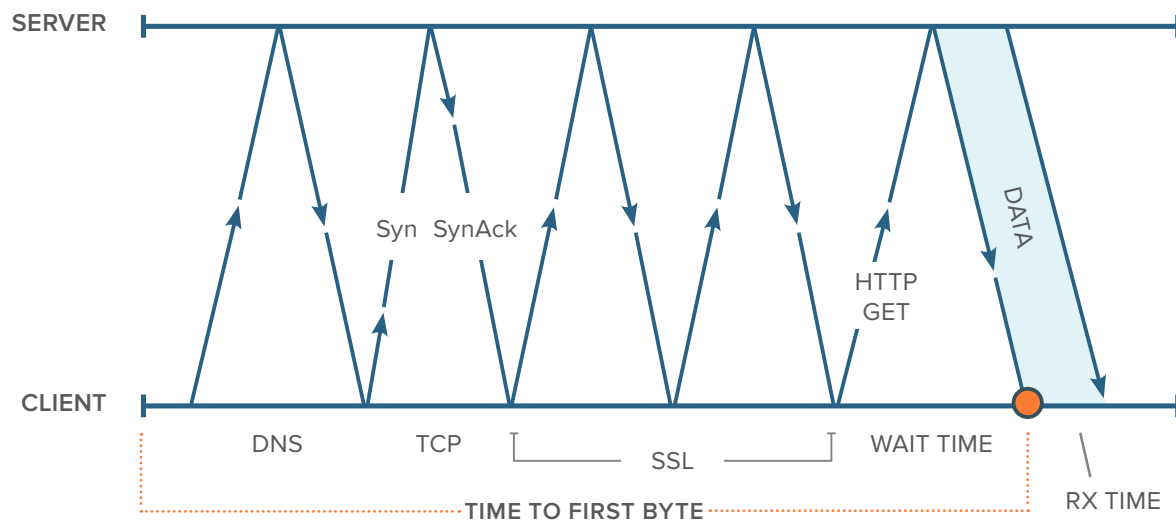*Figure 3:* Application delivery through different network segments.

Figure 4: Decomposing time to first byte (response time). Response time reflects the time for the server to start sending data to the user.

Routing Stability: Routing inside the enterprise is determined by interior routing protocols such as OSPF and IGRP. But for Internet traffic, different networks exchange routing information using the Border Gateway Protocol (BGP). BGP allows independent neighboring networks to communicate and decide what traffic they will exchange. This process is usually driven by economics rather than network efficiency; as a result, routes are not always optimal from a latency or bandwidth point of view. BGP changes (e.g. misconfigurations) can render an entire network unreachable or induce severe performance degradation on applications.

Wireless Access (WLAN): Poor-quality 802.11 wireless access in branch offices is a common cause of application performance degradation. This can be caused by degradation of the radio signal power (lower RSSI) that is originated by interference or physical distance between the client and the access point. This is often a hard element to troubleshoot end-to-end, since typically you can't access this information without instrumenting the client or the access point.

Web Application Performance

Web application performance can be characterized by browser page load traces for a given application. Each page consists of multiple components (such as images, videos, fonts, etc) that are downloaded and processed by the browser. Some of these components originate from third-party sites rather than the SaaS vendor data center. The download process of each component follows a sequence of steps (Figure 4): 1/ DNS lookup, 2/ TCP connect, 3/ SSL negotiation, 4/ HTTP GET (or POST) and 5/ Receive time. Components are typically organized on a waterfall chart, where the x-offset reflects the start of the component's loading time (Figure 5).
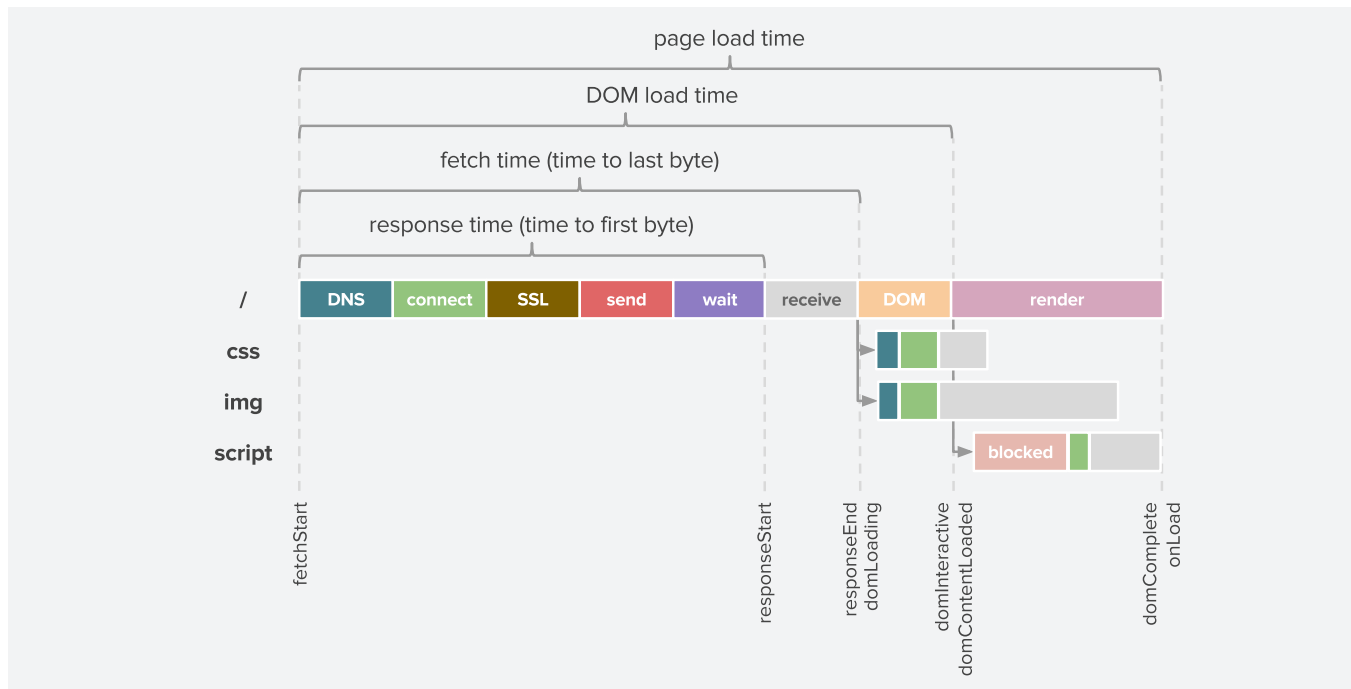
Figure 5: Page load timing metrics and browser events.

There are some key metrics that capture the user perception of how fast a web page loads and interacts:

» **Time-to-first-byte:** DNS lookup time + TCP connect time + receive time; this metric is also called response time and is a good approximation of the time it takes for the server to start sending data to the user. If the web server is not delaying requests, the wait time should be roughly equal to the TCP connect time (and roughly equal to 1xRTT), but in cases where the web server is overloaded this value can be high.

» **Page Load Time:** The time it takes for the browser to trigger the load event for the page. It includes the time to download the HTML, parse the markup, process CSS, render the page and download all images on the page.

» **Transaction Time:** The time between the start step and the end step in a script that simulates user actions in a group of pages (e.g. log in to a site).

SaaS Performance Management: Best Practices

One of the great advantages of SaaS applications is the ability to log in and begin work without the archaic implementation cycles of legacy enterprise software. Performance management of SaaS applications should not be an afterthought. Enterprises should evaluate application performance as part of the vendor evaluation and selection process. This is especially critical when users are spread out across branch offices throughout the enterprise. A SaaS application may have great bells and whistles, but if key users outside of the corporate HQ experience poor performance, those features are of little

> The reality is that the Internet is becoming the corporate backbone for the modern enterprise. Most IT teams may not even be aware of where the applications they rely on are being delivered from.

consequence. Setting up periodic tests that repeat a certain behavior or flow from multiple branch offices is a good way to track application behavior across time and space. Ensuring good performance of SaaS services is largely dependent on close collaboration among the Network Operations team, Application Owners and the SaaS vendor. At a high level, the two aspects to pay attention to are provisioning and monitoring.

Provisioning

» **Routing:** Enterprise architectures may involve traffic backhauling from remote offices via the corporate network and single or limited exit points. Enterprises should ensure that this backhauling will not be a bottleneck for the specific SaaS application. For applications requiring a lot of back and forth with the application servers, it is important to ensure the round-trip delay between a user and the egress point of the corporate network is short. Alternatively if possible, you can use techniques such as split tunneling to allow Internet traffic from branch offices to exit locally.

» **DNS:** Most enterprises will use an internal DNS server and may often block clients from using a third party external DNS resolver. If a user in Singapore uses a centralized resolver in New York, the requests originating from this resolver to the SaaS application could 'fool' the SaaS application into thinking a remote user is in New York instead of Singapore, nullifying any geo-based performance optimizations.

» **Capacity:** Depending on the nature of the application, available network capacity can greatly impact perceived application performance. IT teams need to understand the nature of the SaaS applications and whether bandwidth guarantees are needed to ensure that there is enough available bandwidth inside the corporate network and on transit links. Also consider looking into Quality of Service (QoS) options.

» **Hosting location:** Some SaaS applications are hosted in a specific data center that can be selected by the customer. The selection process should consider the experience of users globally, paying special attention to users in remote markets that may have high latencies to the data center hosting the customer's SaaS instance.

Monitoring

» **Tool Selection:** Once a SaaS application is adopted, whether in evaluation or real use, IT teams should continuously monitor network and infrastructure performance metrics inside the corporate network. With the right tools, IT teams can also expedite the troubleshooting process by using the tools capabilities to identify the root cause. The best practice is to emphasize the troubleshooting aspects of a potential performance management solution before investing in it. Ideally, the selected tools should be able to indicate where the problem is, whether it's in the corporate network, the Internet or the SaaS vendor's data center.

"ThousandEyes has provided us with visibility into global network status that we haven't found with other tools. Easy analysis of BGP state changes and full path visualization tools have routinely reduced our troubleshooting time and resulted in real answers as to what happened."

Steve Loyd
Director of Operations
Zendesk

» **Testing Methodology:** To accurately isolate most problems, the best practice is to test various layers—such as HTTP, DNS and network—separately, but to also be able to correlate between these layers. While passive monitoring techniques are often good for application metrics, active probing is likely the only way to characterize certain aspects of remote network behavior. It's also important to test the application backend to ensure the predictability of responses. Setting up a transaction that can simulate real interaction post login can be a good way to test the data access capabilities beyond simply testing to the index page loading.

» **Problem Resolution:** Once the IT team detects a problem and isolates the cause, fast resolution is always the key to getting things back to normal. With cloud applications, this is easier said than done, since the resolution typically must involve close collaboration among IT teams, Internet Service Providers (ISPs) and the SaaS vendor. Ideally, the monitoring workflow should take into consideration the problem resolution process and streamline it either by building collaborative data sharing capabilities or bringing in monitoring tools that already have this workflow built in.

The Need for Change in Performance Management

Traditionally, enterprises deployed applications in their IT environments as part of a software package, e.g. CRM, E-Mail. When employees complained about performance issues, the immediate response would be to deploy products that measured behavior on the application side, either by instrumenting the application for line of code analysis or by tapping network traffic at the server side. The majority of enterprise performance management products were built for applications that were directly controlled by the enterprise. This is no longer the case.

The reality is that the Internet is becoming the corporate backbone for the modern enterprise. Most IT teams may not even be aware of where the applications they rely on are being delivered from. The average number of network hops enterprise users have to go through has increased drastically, and application delivery has become the key to ensuring good performance. The implication is that users in an enterprise branch office may experience performance problems with a cloud application despite the cloud application having zero issues itself. Add this to the fact that it's impossible for an enterprise to directly instrument third-party cloud applications or parts of the Internet for performance management and troubleshooting, and IT teams now face some very tough performance management challenges. To remedy this, enterprise IT needs to rethink their monitoring portfolio and what it takes to manage performance for cloud applications.

The ThousandEyes Approach

ThousandEyes provides visibility into network and application performance inside enterprise environments and helps IT, operations and network teams locate the root cause of issues in a matter of minutes. ThousandEyes connects each layer of application delivery, making it possible to jump from layer to layer to identify underlying problems. Each layer maps to a different subsystem of application delivery with its own set of unique metrics. ThousandEyes can be deployed using either Enterprise or Cloud Agents to provide broad coverage both within the enterprise and from an external perspective.

Learn more at www.thousandeyes.com.

ThousandEyes

201 Howard Street, 17th Floor
San Francisco, CA 94105
(415) 513-4526

www.thousandeyes.com

About ThousandEyes

ThousandEyes is a network intelligence platform that delivers visibility into every network your organization relies on, enabling you to resolve issues faster, improve application delivery and run your business smoothly.