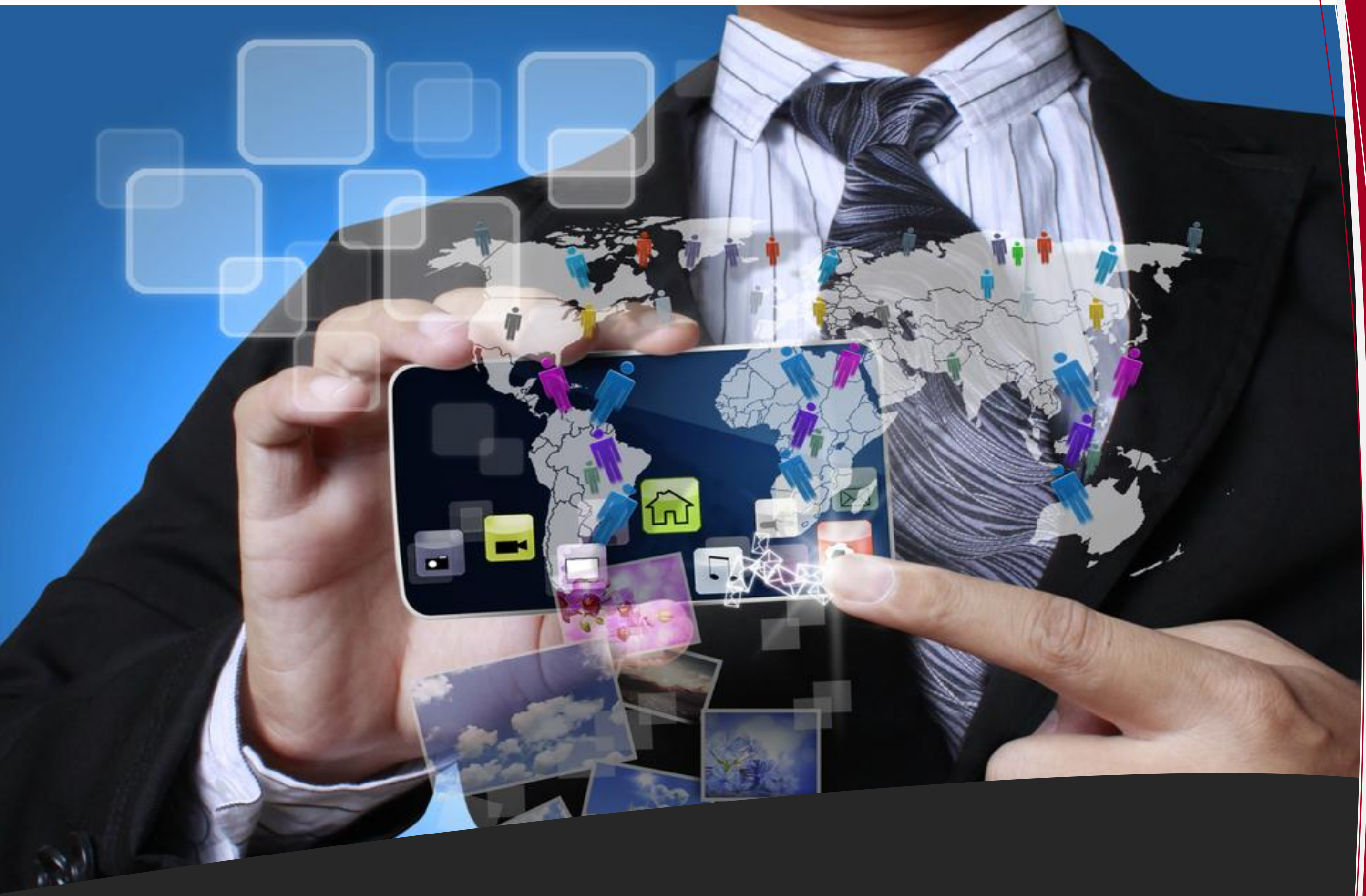


Mobile Application Performance Testing

Aluminum foil, elevators and other mobile testing myths debunked

TABLE OF CONTENTS

Executive Overview **3**

Mobilization is a High-Stakes Game – Play to Win 3

Introduction **4**

“Expert” Advice? 5

Unraveling the Myths 6

The top five Myths of mobile testing **6**

Myth #1 - Testing in the Wild 6

myth #2 - War Driving 7

myth #3 - Freeware and partial emulation (AKA compound inaccuracy) 8

myth #4 – ignore jitter: Streaming Media and full emulation 9

myth #5 – sterile Functional Testing (does a 10 minute RESPONSE PASS?) 10

The Challenges Presented across the Mobile Network **11**

Network Virtualization for Software Testing **12**

An Introduction to Shunra NV 13

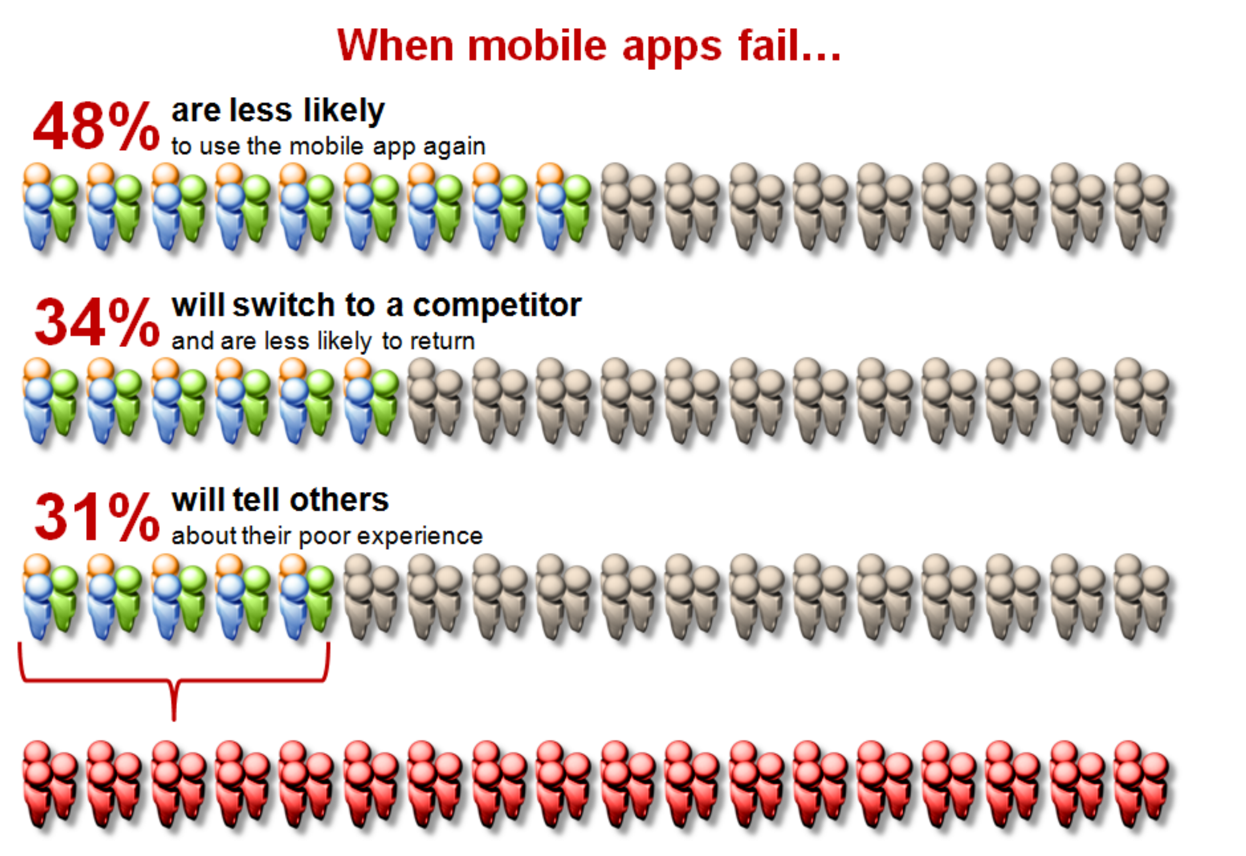
About Shunra **15**

EXECUTIVE OVERVIEW

MOBILIZATION IS A HIGH-STAKES GAME – PLAY TO WIN

Enterprises rely on applications to drive business success, measured in terms of revenue, productivity, customer satisfaction and/or brand perception. As companies continue to adopt a mobile-first approach to application development and deployment, this reliance is undergoing a dramatic shift that changes how businesses interact with their customers and employees.

However, this shift is putting those interactions at risk. Traditional approaches to managing, validating and ensuring the quality of end user interaction have not evolved at the speed of mobility. End users are increasingly faced with slow load times and delays in transaction responses that drive down user adoption and significantly decrease the number of successful interactions.



Equation Research, October 2012

The business risk of performance issues is avoidable if degradations can be detected and remediated before the mobile app is deployed. However, recent research suggests a general lack of mobile app testing experience and awareness of best practices are preventing reliable and accurate testing. At its core, though, mobile app testing requires the same basic tenets of traditional application testing:

- Test environments must accurately reflect real-world conditions.
- Testing must be done under precisely repeatable conditions.
- Testing should include “what if” and edge scenarios.
- Testing must account for performance impact on both users and infrastructure.

These requirements are difficult to achieve due to the dynamic nature of mobility and the variability of the mobile environment: Myriad devices, operating systems, network service providers and network conditions that change based on factors such as time of day, network utilization and distance from cell tower.

While solutions exist that allow testers to account for mobile hardware and even distributed third-party services, the mobile network remains an elusive but necessary component of any test environment. The constantly changing conditions of the mobile network make it particularly challenging, but accounting for the impact of network constraints like latency, bandwidth, jitter and packet loss is essential for accuracy and reliability in load, scalability, capacity, functional and performance testing.

In the absence of proven best practices and methodologies for accurately testing mobile applications, enterprises may employ extreme solutions to validate and ensure performance, but these (often ridiculous) testing myths result only in failed and costly attempts to protect the expected gains in revenue, brand and productivity that come with mobility. There is a better way.

This white paper addresses the challenges and myths of mobile application performance testing, while providing insight into the proven tools and techniques, in particular *network virtualization*, that are requirements to accurately test, validate and optimize mobile application performance. These advanced capabilities are emerging as best practices employed globally by market leaders in both private and public sectors.

INTRODUCTION

As the mobility wave continues to crest – mobile app development is expected to outpace native PC projects by at least 400% in the next several years[i] – development approaches and platforms for mobile are rapidly advancing while approaches to mobile app testing lag well-behind.

The World Quality Report acknowledged the challenges within the software testing market, noting that only 31% of enterprises formally test their mobile apps. Further, over half of enterprises surveyed reported that their QA teams were merely “average” when it came to running effective tests, and most cited a lack of appropriate tools and/or knowledge of testing methodologies.[ii]

This lack of mobile testing experience can cause testers to truly stretch the limits of reality in order to accurately understand how an application will behave. In April 2013, a software quality and testing blog tackled the question of how to test mobile apps. One tester contributed[iii]:

> *You should physically move around to get varying WiFi signal strengths. If you don't want to use your feet, you can begin wrapping aluminum foil around the handset until the signal dies down.*

Over 1.5 million apps are available for download through the Apple App Store and Google Play. Fortunately, approximately 1.5 million tons of aluminum foil are produced annually in the United States and Europe[iv]. In other words, roughly one ton of aluminum foil is available per mobile app for testing and ensuring performance each year, if you subscribe to this approach.

Also in April 2013, an article titled "Losing the Wi-Fi connection: Testing mobile app behavior" illustrated the absurd methods some testers use. In the article, the reader is advised to "just test it" with this recommended approach[v]:

> *My test plan went like this: I opened the application, walked into the elevator, rode it to the basement, then reclosed the doors and tried to use the application.*

The article continues with recommendations like: "Try using the application continuously while a vehicle is moving," and, "For a lightweight approach, move between wireless access points in a large building." The article ironically concludes that the greatest challenge facing mobile app testers is in knowing which of these tests to run first.

Kudos to these testers for knowing that testing the network impact on the application is critical. But, their greatest challenge isn't prioritizing testing efforts. It is in accurately recreating the end user's experience in the test lab.

If all end users were locking themselves in elevators or wrapping their handsets with one ton of aluminum foil, then these tests would be reliably predictive of real-world application performance. Since end users typically have more sense than this, these misguided testing approaches do not deliver reliable results upon which any business decisions should be made.

"EXPERT" ADVICE?

It is difficult to blame testers new to mobile applications for being creative with their methodologies. Only recently have enterprises made the shift from asking how to develop mobile apps to how to test them. These enterprises often turn to the advice of "experts" who are also making the transition from development to testing knowledge.

In a presentation at EclipseCon 2013, a senior industry analyst explained that application teams do not have insight or control over last-mile network conditions. He stated, "It's nearly impossible to replicate actual runtime conditions for all those sources in the lab."[vi]

His solution to this challenge? Fail fast – Fail Often! Deploy the application and test in the wild quickly so performance issues can be found, presumably before actual end users get their hands on the application.

UNRAVELING THE MYTHS

Elevators. Aluminum foil. Testing in production. These myths of mobile performance testing are endemic to the early nature of the mobile performance testing market. They are also contributing to the performance epidemic that has stricken mobile apps. You can see this in the ratings and comments any of us can find in the app stores and in the fact that 94% of development and test professionals surveyed by SOASTA identified significant gaps in mobile testing.

Analysts are quickly coming of age. Forrester Research is now touting the need and ability to discover and emulate production network conditions in the test environment, thereby starting the process of debunking the mobile testing myths.

Other misinformation exists, however, that also threatens the revenue, productivity and brand gains that mobile apps offer to the enterprise.

THE TOP FIVE MYTHS OF MOBILE TESTING

Aluminum foil and elevators are obviously not adequate approaches to account for the network layer that is often missing in mobile app testing. However, there are other commonly accepted approaches that, while inadequate and unreliable, are nonetheless put into practice today.

MYTH #1 - TESTING IN THE WILD

Perhaps the most common and risky approach to testing performance is to simply deploy the mobile app and let it be tested in production. “In the wild” testing means performance data will be based on real users and real network conditions. But this approach poses significant threat to real end user experience and business value.

Desired end user engagement or action is at risk if end users are presented with a disappointing experience that causes them to abandon the mobile application or site and perhaps visit a competitor’s app. In addition, performance remediation and optimization in this scenario only occurs if end users: report the problem, accurately relay the exact steps they took to create the error and note the conditions under which the application was being used.

> *End users are not testers. Using them as testers means they will be the first to experience performance issues.*

Customers typically do not do this with the reliability and certainty required to effectively troubleshoot and remediate an issue.

Crowdsourcing "in the wild" testing is just as risky, if not more so. By employing testers, sometimes globally, to simultaneously test your in-production application, you would expect to be able to understand how load and network conditions affect application behavior.

While crowdsourcing degrades "real" user experience as a result of creating unnatural congestion on the network, the most significant disadvantages of this myth are the inability to cover all real-world usage scenarios and an inability to collect the necessary data for remediation.

Application behavior measured under crowdsourced testing in the wild is therefore not accurate or predictive of actual application performance, does not facilitate rapid issue resolution and poses risk for real customers or end users trying to access your systems at the same time as "testing."

Recreating error conditions in a controlled, repeatable manner for analysis, diagnosis and remediation is challenging at best when relying on globally distributed, crowdsourced testers all operating in unique and varied environments. This makes problem resolution timely, costly and imprecise, as the root cause of the issue may not be correctly identified.

MYTH #2 - WAR DRIVING

Similar to testing in the wild, some development and QA teams may test an application by only deploying to a few known testers and having those testers walk around a building or drive around a city in order to evaluate how an application will perform under varying conditions, including network hand-offs and poor signals.

While this approach poses less risk to real users, it offers no consideration for the production network conditions experienced by the real distributed user base. Rather, it provides a test that reveals performance data only for a particular location at a particular time of day. It is not adequately representative of distributed user populations experiencing varied network conditions. In addition, performance issues encountered while "war driving" are typically difficult to repeat and recreate, making remediation costly and imprecise.

Both testing in the wild and war driving are inherently flawed approaches to ensuring performance because they make use of the production network and are tactics employed after applications are deployed. These options for testing in production are not ideal because:

- Real end user experience is threatened.
- Performance issues are not easily captured, making diagnosis and remediation challenging.
- Errors that are isolated are more expensive to fix than those caught early in the development cycle.
- Remediation is up to 100 times more expensive for issues discovered in production as opposed to earlier in the software development life cycle.

Testing in the wild and war driving put revenue, productivity and brand at risk by extending mean time to resolution and requiring remediation to take place at the most costly stages of the application life cycle.

MYTH #3 - FREEWARE AND PARTIAL EMULATION (AKA COMPOUND INACCURACY)

Fortunately, technologies exist to test performance earlier in the application life cycle, when testing is more cost-effective and efficient. By thoroughly testing before deployment, potential issues can be identified and remediated before end users and the business are affected.

Even if you understand the requirement to recreate network conditions in the test lab, some solutions provide a less-than-complete attempt at network virtualization. Some test automation solutions introduce the concept of bandwidth as a network constraint. This represents a fractional first step when it comes to load testing, for example, where different user populations may experience different bandwidth conditions, for example 3G versus 4G mobile networks.

Often, bandwidth limitations are simply static, rather than dynamic, settings. Static bandwidth makes it impossible to recreate the experience of a mobile user who may be experiencing varying signal strength on a single network, switching between networks (4G to 3G for example) or moving from WiFi to a 3G connection. Further, if the bandwidth metric, or any network condition virtualized in the test lab, is not based on actual measured conditions from the production network, then it represents an additional variable of inaccuracy that introduces additional risk and uncertainty to test result reliability.

If static bandwidth were the extent of the risk presented by limited emulation, then the argument could be made that simply testing edge cases with (low) bandwidth would be sufficient. But, contrary to popular belief, bandwidth is not the be-all-end-all of performance.

When it comes to mobility and application performance, latency rules supreme.

The amount of time it takes data to move from one endpoint to another is the largest contributor to application performance. Roundtrip time for requests and the impact of handshakes between routers, within the cloud and other network equipment contribute to latency. So do coding techniques and network protocols that govern how and when data is sent.

Similarly, you must account for jitter or variability in latency. Jitter can be difficult to represent in a test environment. The static variables for bandwidth or latency are easier to create. As a result, some solutions downplay the value of jitter.

MYTH #4 – IGNORE JITTER: STREAMING MEDIA AND FULL EMULATION

It is dangerous to ignore jitter, particularly in the case of streaming media.

Consider the fact that as networks continue to evolve and 4G (and future) technologies take hold, the amount of streaming content delivered to mobile devices will rise exponentially. By 2017, over 20,000 petabytes of data traffic will be sent over mobile; streaming video will be the primary driver of growth for mobile data traffic.[vii]

The Challenge of Mobile VoIP and Video

The digital revolution, coupled with advances in mobility, has changed the way we live, work and learn. Streaming content like video can now be customized and personalized; content providers have options for new rich media formats that promise broader reach and deeper customer engagements.

These providers are in a constant struggle to meet increasing customer demand for advanced video services, anytime/anywhere access, availability across mobile terminals and screen sizes, telephony and faster Internet speeds, all while trying to manage the cost and maintenance of complex networks.

Linear applications such as VoIP and video drive significant real-time traffic onto networks, and are intermixed with non-real-time traffic such as email or other back office applications.

For reliable VoIP and video transmissions, packets cannot be dropped, re-ordered or experience jitter. Codecs with error resiliency can help, but developers need to plan and test for appropriate compression and decompression over multiple user networks to understand what the delay will be, how errors will be handled (gracefully or not) and what the viewer's experience will be.

Streaming media is particularly susceptible to jitter – data packets must arrive to the end user device in the correct order and in a timely manner, otherwise playback will be choppy and inconsistent.

A customer-centric view on experience, or streaming content performance, is therefore paramount. It follows that both network and application performance must be adequately tested and managed to deliver satisfactory performance to the end user. Most current tool sets and methodologies fail to bridge the divide between network performance and the performance of applications over that same network. This gap means operators don't have sufficient visibility into a customer's app experience.

Issues can manifest as frozen frames, choppy videos, low-quality resolution and long buffering delays, all contributing to customer dissatisfaction, service abandonment and revenue loss. Video networks are a composite hybrid of satellite distribution systems, legacy infrastructure and futuristic fiber optics. Performance varies widely depending on a number of factors including service provider, network type, end-user location, time of day, number of users and even weather. Mobility adds yet another layer to this equation as content moves between a constantly changing network of wireless, 2.5G, 3G, 4G and/or LTE conditions.

To manage this burden, streaming content providers cannot rely on static bandwidth or partial network emulation alone. Network conditions such as latency, limited bandwidth, packet loss and jitter are all critical factors that must be taken into account when testing or validating how an end user will experience streaming content delivered over any network.

However, capturing and emulating in a test lab the network impairments that will impact application performance can be a daunting task. In the absence of these conditions, testing will yield unreliable results. The effect of network constraints on end user experience, as well as communication with external services, will be unaccounted for.

Simply adding “think time” to a test script, or using the bandwidth option with an automation tool, will cause a transaction to operate at a slower pace. But fixed transaction delay does not represent real world conditions or the variation of those conditions over time.
In the real world, remote end user transactions cause contention for back-end resources and introduce bottlenecks along the network path. Geographic distance and load on the network influence transaction response times. These conditions cannot be recreated with static values and must be based on real-world discovery in order to ensure accurate and reliable network virtualization for software testing.

MYTH #5 – STERILE FUNCTIONAL TESTING (DOES A 10 MINUTE RESPONSE PASS?)

The term “performance” is sometimes misused by testing vendors. A limited use of network conditions in a single user or functional test does not adequately represent how distributed user groups will experience app behavior.

It is important to differentiate functional from performance testing. How an application responds to a command (user input) is a functional consideration. How quickly the application responds is a performance consideration.

Functional testing can be executed on real or emulated devices. Regardless, functional testing must be paired with performance considerations in order to deliver reliable results data. Development and test teams should be wary of misinformation and confusing marketing messages that promise an “end-to-end” solution. These could leave an application at risk if they don’t include consideration for the network.

To gauge how “end-to-end” a solution really is, development and test teams should ask the following:

- ➢ What network conditions are virtualized?
- ➢ How are those conditions captured or discovered (are they from the real network)?

- Are they virtualized to represent different user groups or only for a single user test?
- Regardless of the amount of load being applied, are multiple network conditions being virtualized to represent distributed user groups?

If pitched an “end-to-end” mobility testing solution, the last bullet above is critical. As with limited emulation, the full scope of network constraints and the ability to virtualize different sets of constraints for different user groups is typically a missing element in the “end-to-end” functional testing pitch. In the absence of this capability, “end-to-end” is a misleading sales term that covers up the risk of unreliable performance test results.

Some vendors attempt to overcome the challenge of network virtualization by offering functional testing in the cloud. While conditions between the test cloud and the tester are accurately created and their impact accounted for in a test, real end users are not likely in the same location and accessing the application over the same network as the tester.

Testing in the cloud provides flexibility and insight into performance for one particular location. It is not representative of the total user base. It is also dangerously inaccurate as tests in the cloud enjoy substantially faster networks than real users at home or in most enterprises.

Functional and load testing are important elements in an end-to-end solution. However, both of these tests require the ability to emulate multiple network locations and users in order to deliver accurate and reliable insight into how an application will perform once deployed to a distributed population.

THE CHALLENGES PRESENTED ACROSS THE MOBILE NETWORK

If developing and deploying apps that support myriad devices and operating systems is difficult, then accounting for the network conditions that impact communication between end users, an application and its dependencies is a Herculean task in the absence of the appropriate knowledge and technologies.

Up to 70% of an end user’s experience with a mobile app is dependent on network conditions. In a blog post for monitoring company Boundary, Enterprise Management Associates research director, Julie Craig, writes:[viii]

> *I have long been a proponent of the network as the single most important instrumentation point for tracking and monitoring applications. From my perspective, this network-centric perspective is the single source of truth….*

A network-centric approach to the software development life cycle is critical. While the network itself is not often the root cause of a performance issue, developers, testers and operations teams must consider the network impact in all phases of their efforts. Applications must be designed, developed, tested and deployed with the network in mind.

For example, last mile conditions are hyper-dynamic due to factors like time of day, utilization, distance from tower, number of bars and hand-offs between networks. Beyond the tower, network connections between the application and third-party services, content delivery networks (CDNs) and cloud infrastructure create complexity that also impacts application performance. Enterprises must engage in a network-aware approach in order to guard against performance degradation that could arise from network-related coding and configuration issues.

The distributed nature of applications and end users, along with increasing reliance on cloud and mobile delivery, means end users, whether they are aware of it or not, are more reliant than ever on the network connections that exist from their device to the tower and back through the application infrastructure. All of these connections are constrained by latency, jitter, bandwidth and packet loss conditions.

Accounting for these conditions and reliably incorporating the “network layer” into mobile app testing remains a missing element in the testing process.

NETWORK VIRTUALIZATION FOR SOFTWARE TESTING

For developers and testing teams who are deploying mobile applications, “just test it” means accounting for the network conditions that exist and recreating those conditions with precision. Other methods have been tried, from the ridiculous (aluminum foil and elevators), to the sublime (promises of end-to-end testing with functional solutions). Yet the state of mobile application performance continues to deteriorate.

The missing network layer, while elusive in test, is available with a proven reliable approach termed “network virtualization.”

Network conditions such as latency, limited bandwidth, packet loss and jitter are all critical factors that must be taken into account when testing or validating application performance. This is especially true when considering the multiple network connections required to support third party services and external resources.

> *The missing network layer, while elusive in test, is available with a proven reliable approach termed “**network virtualization**.”*

It is challenging and costly to bring real-world services into the test environment. It is also a daunting task to incorporate the real-world networks affecting these services and the end users into a test environment isolated behind a firewall. The global mobile network, for example, is a communication channel that is in a continual state of flux, with dynamic conditions dictated on a moment-by-moment basis according to carrier, connection type, connection strength, time of day, location and myriad other factors.

However, without virtualized network conditions that account for the dynamic and complex nature of mobility, test results are not reliable. Communications between the end user, the application and the application's dependencies in test will not represent the real world.

With the massive global uptake in mobile devices and the rise of cloud computing, the effect of the network has become even more pronounced. Conditions across the "last mile" affect performance more than other factors. Mobile conditions, in particular, are intrinsically dynamic. Therefore, incorporating network virtualization is particularly crucial when any mobile or cloud component is part of the scenario.

Businesses must test for and ensure end user experience. They must adopt a comprehensive network virtualization strategy that includes:

1. **DISCOVERY**
   Identify and record actual application infrastructure and production network conditions, business processes, application topology and deployment scenarios. Know and capture the environmental conditions affecting your end users' experience.

2. **VIRTUALIZATION AND TESTING**
   Virtualize those production environment conditions, including users, services and networks in order to accurately emulate user experience. Integration with automation, load and functional testing tools enables reliable single-user and multi-site/multi-user testing.

3. **ANALYSIS AND OPTIMIZATION**
   Thoroughly analyze test results to identify potential performance bottlenecks, validate performance and ensure service level objectives (SLOs) compliance before application deployment. Identify and implement optimization strategies to improve performance and ensure a positive end user experience before deployment.

AN INTRODUCTION TO SHUNRA NV

Shunra NV is the most accurate way to recreate production network conditions in a pre-deployment lab. The suite of products provides support for capturing and emulating network impairments like bandwidth constraints, latency, jitter and packet loss. In addition, Shunra NV enables simulation of multiple locations and user groups in the test lab. This is essential for accurately recreating production-level environments with mobile and Internet users.

The first step required to implement network virtualization is to "discover" the real-world network conditions affecting end users and application services. Shunra NetworkCatcher unobtrusively captures production network constraints like bandwidth, latency, jitter and packet loss. Recorded interval statistics can then be re-played, or virtualized, in the test environment, using the Shunra NV suite of products, including Shunra NV Appliances, Shunra NV Desktop, Shunra NV for Mobile and Shunra NV for HP Software.

With Shunra NV, virtualized network impairments, such as high jitter rate and limited available bandwidth, can be emulated in the test lab so that the effect of the network on transaction response time can be reliably and accurately measured.

In addition to capturing real-world network conditions, Shunra's network virtualization solutions for software testing include a Global Library of mobile and broadband network conditions. This library, built on millions of data points, provides typical, best and worst case conditions recorded from major cities around the world.

With Shunra's network virtualization technologies, custom impairments can be applied to determine the specific effect on application and network behavior in order to emulate worst case scenarios or to troubleshoot specific in-production problems. Shunra's technology also provides the ability to execute tests, analyze results and remediate or optimize performance.

[i] "The Year of the Enterprise Tablet," Vertic. April 18, 2012. http://www.vertic.com/blog/year_of_the_enterprise_tablet_infographic/

[ii] "World Quality Report 2012-2013," Capgemini. September 2012.

[iii] "How to test mobile application which uses wifi connection," Software Quality Assurance & Testing. April 5, 2013. http://sqa.stackexchange.com/questions/5884/how-to-test-mobile-application-which-uses-wifi-connection

[iv] "Aluminum foil," Wikipedia. http://en.wikipedia.org/wiki/Aluminium_foil

[v] "Losing the Wi-Fi connection: Testing mobile app behavior," by Matthew Heusser. SearchSoftwareQuality. April 15, 2013. http://searchsoftwarequality.techtarget.com/tip/Losing-the-Wi-Fi-connection-Testing-mobile-app-behavior

[vi] http://searchsoftwarequality.techtarget.com/tip/Forrester-VP-says-enterprise-mobile-applications-miss-the-mark

[vii] "Mobile Data Traffic to Grow 300% Globally By 2017 Led By Video, Web Use, Says Strategy Analytics," TechCrunch. July 3, 2013. http://techcrunch.com/2013/07/03/mobile-data-use-to-grow-300-globally-by-2017-led-by-video-web-traffic-says-strategy-analytics/

[viii] "Application Performance in a 'Cloudy' World," by Julie Craig. Boundary. February 11, 2013. http://boundary.com/blog/2013/02/11/application-performance-in-a-cloudy-world-2/

ABOUT SHUNRA

Shunra delivers a proactive approach to application performance engineering with network virtualization for software testing at its foundation. Shunra NV™ solutions account for the challenges of real-world network impairments (WAN, Web, Mobile, Cloud) into all business and mission-critical applications prior to deployment.

An independent survey from TechValidate reports Shunra customers experience up to a 70% reduction in production performance remediations and a decrease in application delivery cycles of up to 40%.

Shunra is the industry-recognized leader in network virtualization for software testing, offering over a decade of experience with some of the most complex and sophisticated networks in the world. Shunra has over 2500 customers worldwide, including 75 of the Fortune 100. Customers include Apple, AT&T, Bank of America, Best Buy, Boeing, Cisco Systems, Citibank, eBay, FedEx, GE, ING Direct, Intel, Marriott, MasterCard, McDonalds, Merrill Lynch, Motorola, Oracle, Pepsi, Pfizer, Siemens, Target, Thomson Reuters, Verizon, Walt Disney and the U.S. Federal Reserve Bank.

www.shunra.com
215.564.4046