Spirent™

WHITE PAPER

Critical Test Patterns and Measurements for SASE and SD-WAN Environments

Overview

The accelerating adoption of Secure Access Service Edge (SASE) and Software-Defined Wide Area Network (SD-WAN) environments has great potential for enhancing security and performance while cutting costs compared to traditional Multi-Protocol Label Switching (MPLS) or dedicated point-to-point lines. However, modern virtualized architectures also increase the complexity of testing, introducing new risks. It is increasingly critical to test security-related patterns accurately and realistically and to define meaningful measurements for both SASE and SD-WAN environments.

Here are some of the unique testing challenges this paper will address:

- **Defining Traffic Patterns:** We must redefine the minimum test traffic pattern to represent normal daily traffic, as observed in the traditional WAN compared to exclusive simple, mono-sized objects. Stateless traffic and simple stateful traffic, cloud applications and Zero Trust (ZT) policies must be properly positioned in the test plan to provide testing value.
- **Defining What to Measure:** We must define what to measure and how to measure it so that we can accurately predict real-world traffic concurrency within SD-WAN infrastructure and SASE security principles that work on top of any infrastructure.
- **Defining Test Load and Duration:** SD-WAN and SASE change how we load and sustain traffic over a testing iteration. We must update our load and duration parameters to match the realities of the SASE/SD-WAN environment.
- **Defining Service Mix and its Impact on SD-WAN:** The distribution of traffic flowing over an SD-WAN link and SASE infrastructure can alter the behavior of the link. We must define a meaningful distribution mix.
- **Effects of Physical WAN on SD-WAN Bearing Capacity:** How does inherent physical attributes of WAN such as distance, latency, and drop effect scale?

Traffic Pattern is Core to SASE and SD-WAN Testing Success

The unit of traffic pattern that we select for testing SASE and SD-WAN is absolutely critical to a meaningful and rigorous test process. To fully understand why we must use more complicated and realistic traffic patterns compared to a simple L3 iMix or simple fixed size HTTP, we must explore what is different in SASE/SD-WAN environments compared to fixed, private leased line or MPLS-based WAN solutions.

The first consideration is **security**. You can assume that your confidential data is being monitored, recorded, parsed, and stored without your knowledge by organizations with very deep, state-sponsored resources. In some cases, your organization can become liable for data breaches, and incorrect deployment can place your organization in jeopardy.

Some SASE and SD-WAN infrastructure providers will offer security as a service, basically offloading security to a third party. But any time you relinquish control of security, you increase likelihood of a "security event." With all of these variabilities and unknowns, how can we start to create a meaningful and comprehensive test plan?

For all SASE scenarios where the traffic is being processed by some upper layer node like a firewall, simple HTTP/HTTPS concurrent connections, bandwidth, and CPS rate test classes are a mandatory part of the scope of testing. An additional consideration that SASE brings to the testing sets is Zero Trust principles and granular access control to specific applications based on the user context. Business application emulation and L7 traffic load generation is crucial for testing of all baseline capabilities of SASE Zero Trust Network Access (ZTNA), Cloud Access Security Brokers (CASB), Data Loss Prevention (DLP), Secure Web Gateways (SWG) and NextGen Firewalls (NGFW).

There is another very fundamental problem with using these test patterns if scalability is inferred. Because of so much dynamic variability and underlying conditions that you may not be aware of, these patterns tend to be overly optimistic and can imply capabilities and performance levels that are not achievable in a real-world scenario. False-positive results must be avoided.

Correctly forming a test traffic unit will help us adapt to the variability of the SASE/SD-WAN environment.

In traditional WAN environments, the underlay infrastructure is demonstrably predictable. This class of network tends to have fixed routing and tunneling infrastructure and fixed routing paths. For example, in an MPLS network a core "P" router tends to use high performance ASICs and a fairly stable and converged routing table. This path is typically tested with labeled L3 traffic and core QoS metrics such as loss, bandwidth, latency, and jitter. Even under failure conditions where pathed traffic is rerouted, convergence times are generally non-detectable by upper layer stacks, such as TCP, if it does not exceed 200 ms. For example, a 50 mSec convergence time due to MPLS path failure is negligible to TCP timers and upper layer protocols. Even unicast routing protocol failure would be caught by BFD within milliseconds as opposed to a more classic seconds timeout.

When you examine a traditional WAN link, you have a very robust, semi-private, measurably predictable link with years of proof of service. With classic WAN links, you gain a measure of predictability in your application Quality of Experience (QoE) because you can normalize out the underlay of the WAN as a constant variable. Furthermore, traditional WAN circuits are effective pseudowires, not providing higher-level data processing. The downside is that you pay quite a bit for this quality in circuit fees, deployment time, and modern high-value features.

An SD-WAN environment swaps out the predictable MPLS underlay with an "unknown of unknowns" scenario. SD-WAN environments are both layered and compound in nature. For example, the last mile of an SD-WAN link may be an IPSEC tunnel over the local provider's infrastructure for cost-saving purposes.

With the SASE principals on top, traffic could be blocked or steered based on the user and device access context, and new service chains could be created on-flight and be unique to the user level. The next hop may enter a hypervisor and be processed by content-aware switching, and so forth down the chain.

In this case, all the levers of disruptions that virtualization presents to VNF networks and NFV devices may impact traffic. For example, the NFV content switch may ride on Linux, which is using virtualized memory compounded by the hypervisor, which is also virtualizing memory. This double virtualization of RAM, which may in some cases be mitigated by container approach such as with Docker or LXC, has as a potential to insert non-predictable jitter within the transaction flows. CPU core sharing and NUMA node consideration will likely have a randomizing effect on CPU performance. Pinning the cores and RAM may help stabilize some aspect of this variability, but this rapidly cuts into the ROI value proposition of using virtualization.

SASE and SD-WAN links are complex blends of technologies. The weakest link will break quality.

Furthermore, we must recognize that NFV devices are very new compared to physical equivalents. The substantial proof of correctness physical WANs can demonstrate because of actual field deployment cycles are not present in SD-WAN devices. In addition, the conversion of libraries deeply reliant on ASIC and specialized hardware with ultra-predictable timing and performance to x86 equivalents reduce confidence. Years of testing may not be applied to the NFV equivalent. They must be tested from the beginning, and over time and real-world processing years of service. SD-WAN paths may also include physical devices or blends of physical and virtual devices. Because SD-WAN is a "weakest link" class technology, traffic hopping from one class of device to another may be broken by the weakest point or compound effects of hops.

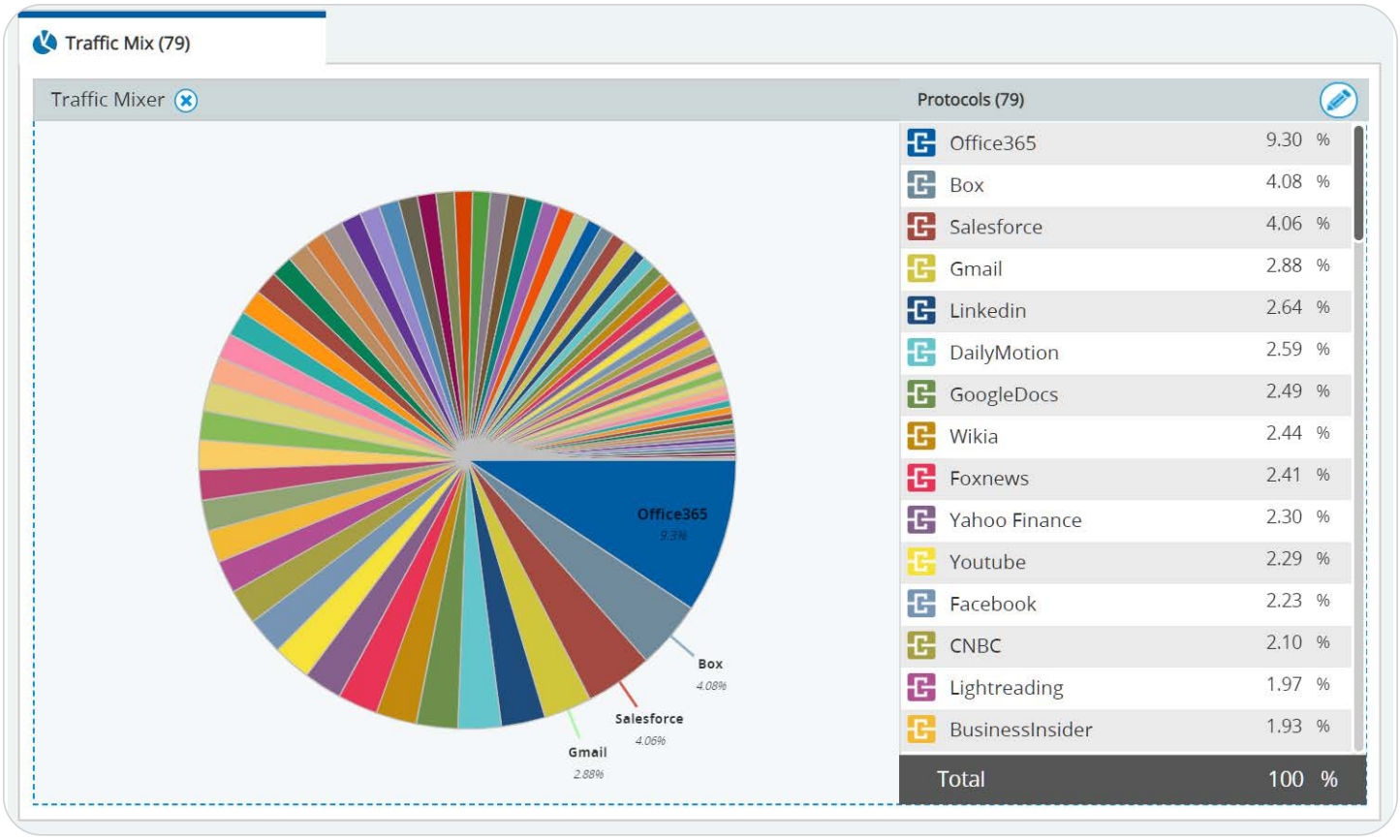
SD-WAN is far more than an emulated wire. Whereas an MPLS router looks at the label and forwards, depending upon the policy the WAN may route traffic or transform traffic deeply within the content of the flow. For example, an SD-WAN policy may route a specific user to a specific part of a CRM-based application at a specific time and location. This is very different from label forwarding. The amount of compute capacity and intelligence in the network has a strong impact on traffic patterns.

SASE and SD-WAN stateful processing of content fundamentally changes test traffic patterns.

Classic measurement of bandwidth as a primary KPI suddenly becomes not sufficient to measure performance, because it will tend to measure best-case scenarios rather than real-world scenarios. The measurement we take must go substantially longer and deeper than classic bandwidth measurements allow.

Let's take a moment to understand what you are swapping out from the classic MPLS network when your SD-WAN circuit rides over an internet-based or semi-public SD-WAN.

First, you lose most forms of assured **Quality of Service (QoS)**. Unless your SD-WAN provider has "fast lanes" across their network, your critical WAN traffic will ride next to video, BitTorrent, gaming, and other dynamically loading applications. Not only do you not know instantaneously the QoS foundation of the traffic flowing over your SD-WAN; you have no way of knowing how QoS will change moment to moment, with no future assurance of quality without proper patterning.



In many ways, your critical WAN links are built on a foundation of application quicksand. You have no assurance of a fixed path of the SD-WAN. Internet traffic routing can change on the fly, adding latency, jitter, and pathing through possibly oversubscribed routers or even satellites.

Next, it is unlikely that all your SD-WAN links will flow through a single internet provider. More likely, your traffic will traverse congested peering point. These tend to be points of random congestion based on time of day. Even though you may have a QoS agreement with an ISP, QoS is only as good as the first hop failure, and may break down after peering, because QoS does not necessarily extend to third party ISPs.

What traffic patterns should be used before testing at an Application and QoE Level? Traditional L3 iMix stateless traffic should be considered a pre-engineering peek forwarding scale test. It will tell you the upper bounds of performance ceiling and will only tell you by frame size what the peek forwarding rate over the test iteration.

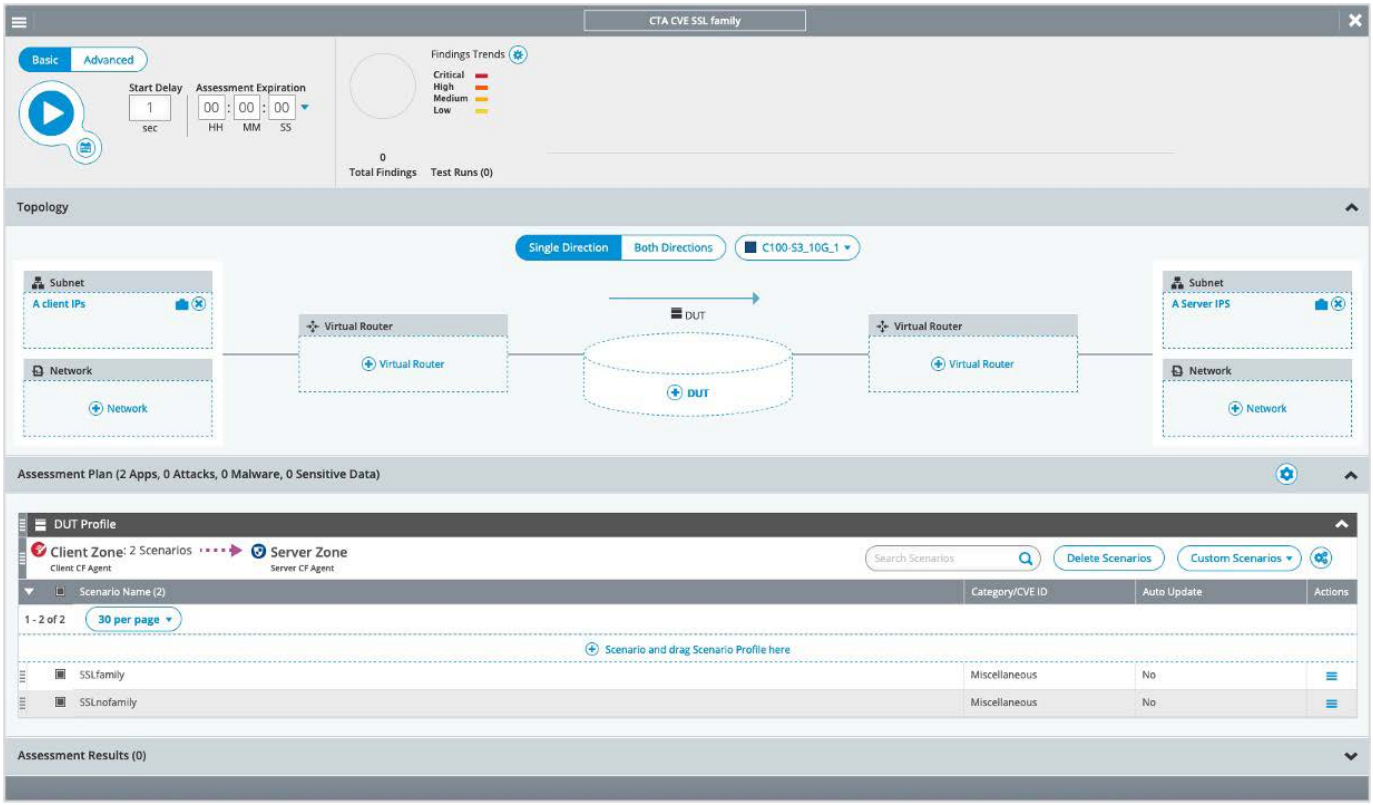
Traditional QoS metrics like latency, loss and jitter will only give you a rough estimate of what to expect because the underlay internet link is always changing, but are important first steps to understanding the behavior of the circuit. The danger of this traffic pattern is misinterpretation of results.

Although peak performance metrics are necessary, it is also not sufficient to imply baring capability of the SD-WAN tunnel. This form of pattern will find gross violations of performance such as packet loss and high latency variations that would certainly affect customer bearer traffic. If it fails at L3, there is no point to test higher up the stack.

QoS metrics are bound together in a policy per QoS differentiated service level. Best practice would be to first define QoS levels (Gold, Silver, Best Effort (BE)) and then for each level to write a logic statement that describes the minimally acceptable level for each KPI (EX. Max Latency < 20 mSec & Jitter < 3 mSec < No Packet Loss for Gold).

When there is contention between QoS levels, it is important that the right level is being prioritized. If Gold is instantaneously competing for resources with Silver or BE, prioritize in this order: Gold first, Silver second, BE third.

Finally, Transmit realism must be addressed. On a per QoS level (Gold, Silver, BE), the tester must have a frame size distribution that matches what is observer in the production networks as well as a loading pattern that is observed in the production network, both generally acquired from inline network data brokers.

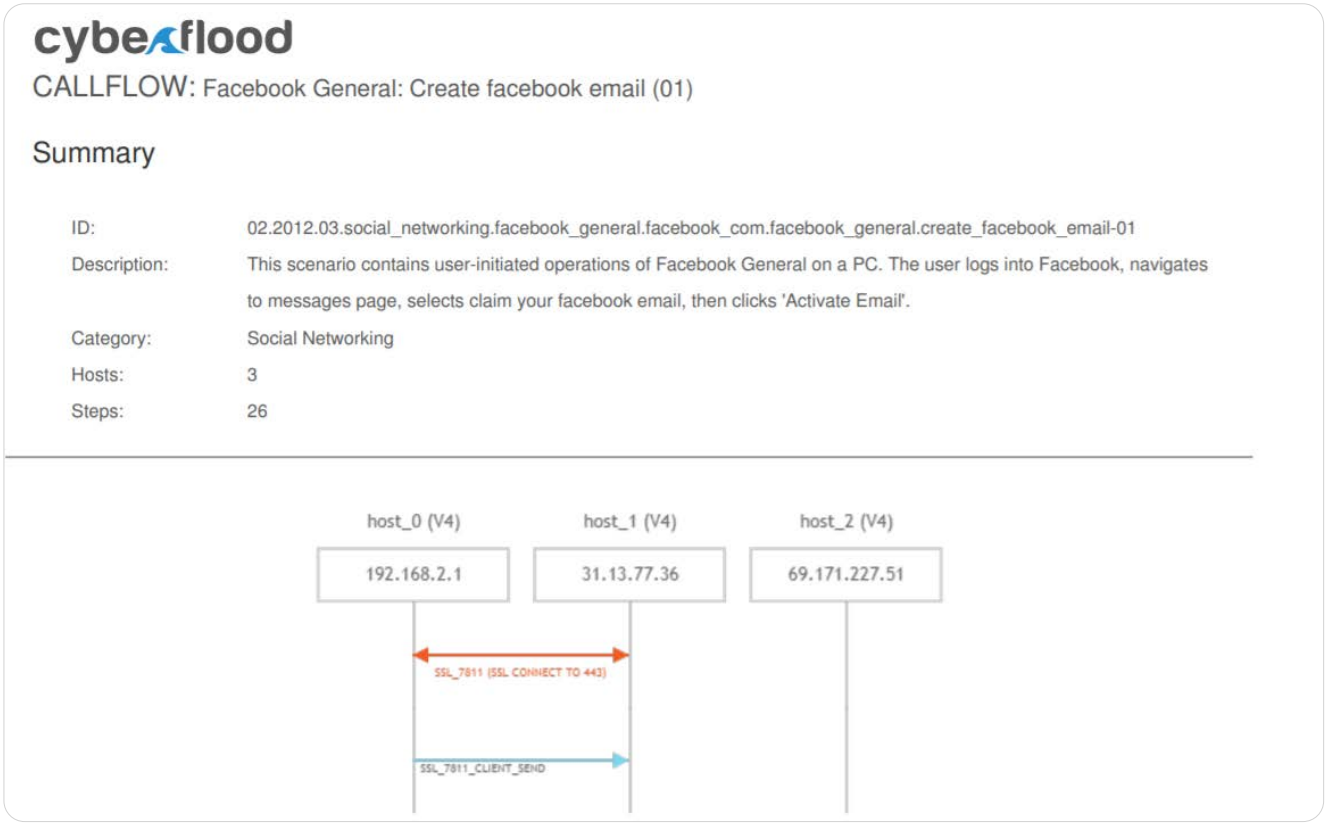


So, what is the meaningful traffic pattern to use for our testing unit? We must go back to our classic MPLS WAN and examine how traffic is generated and observed. The first observation is the predominance of HTTPS services. Traffic on a WAN generally consists of mission-critical enterprise web-based applications (e.g. CRM system, order entry, WebEx, etc.) as well as key applications (e.g. Exchange, VoIP, etc.) mixed in with generic internet services (e.g. Facebook, Skype, etc.).

In most cases, the mission-critical services are the highest priority since most organizations depend on predictable, uninterrupted access to these services. General internet traffic for most organizations is offered as a best effort service. In most cases with SASE/SD-WAN, only internal traffic would be routed though the SD-WAN tunnel anyway, so we can bypass non-critical best-effort traffic unless there is an outbound DPI content scanning security policy for sensitive information. Second, since internal servers and datacenter impairments would be the same for MPLS classic networks and SD-WAN links, we can ignore server congestion as a cofactor in evaluation.

If we focus on the critical HTTPS services, we then have a good model of a basic test unit. This basic unit should be enough bundled traffic to form the service, since people interface with services over the SD-WAN circuit. In addition, it should be measurable by the end user as a unit. For example, if a 100K object over an HTTPS link was tested in isolation this would fail our test because in the real network, that object is bound to many other objects and users cannot qualitatively detangle from the performance of a service. The correct level the is the HTTPS page (e.g. CRM homepage). The modern page is between 150 and 200 URLs, will be bound by rules of HTTP persistence and pipelining, encrypted and is a measurable unit by the end user.

The other critical advantage of aligning the test page to a real-world page with the same level of depth and complexity in a 1:1 ratio, is that we greatly nominalize out false positives in our testing. Since we are generating in the test network the same pattern that will be used in the production network, we are directly testing with patterns the user experiences. Any combination of co-factors in the SASE/SD-WAN chain will be correctly aligned and measured giving us specific, targeted measurement of real-world performance.



Applications and Service Mix Distribution

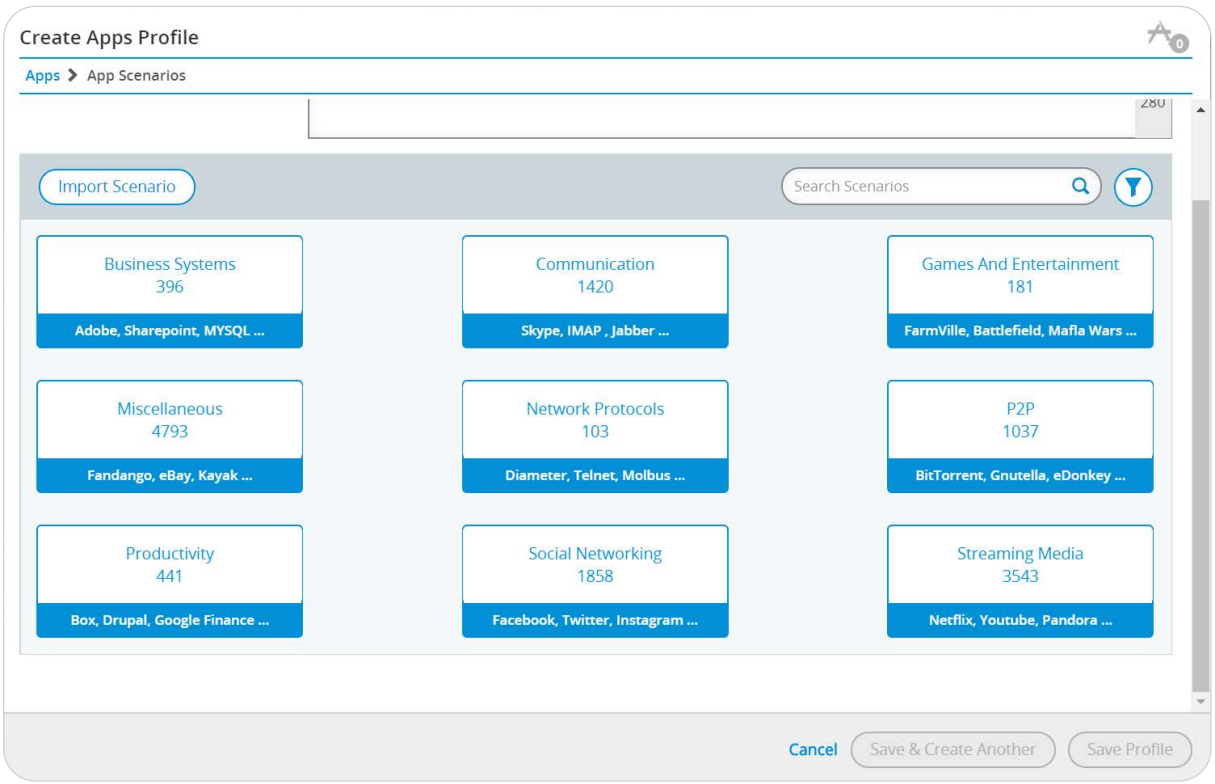
The distribution of services concurrently flowing over an SD-WAN link is critical to define and use when properly testing the circuit. Moreover, when the SASE or SD-WAN environment is enhanced with value-added services such as WAN acceleration, DPI, or IPS/IDS services, having a realistic mix of traffic will meaningfully load the elements in the WAN. We must look at what is generated with a meaningful mix of traffic.

First, HTTPS-based services cause an enhanced SSL compute load on the mid-span SASE service, especially if DPI inspection is enabled. At a minimum, the service will inspect SSL/TLS SNI information to pass/no-pass traffic in the circuit. If the service is terminating SSL and thus an SSL endpoint, additional SSL tunnel management will be stressed. Interior to the device, deep packet inspection will probably occur as well. At lower layers, independent TCP tunnels may be dynamically inspected and at even lower layers, a realistic mix of frame sizes will be generated, changing with time.

Because users may be sharing the same SD-WAN tunnel, or traffic may be concurrently inspected by a single point of failure, the concurrency of multiple services will provide additional stress on the SD-WAN infrastructure.

It is typical for the number of services to be flowing across a WAN link to be in the 10-30 unique services range. In the real SD-WAN network, these services are independent of each other and are randomly generated, changing loading characteristics over time, which potentially adds an infinite number of test cases. This becomes untestable, so we "test to the worst case" which is when all services are concurrently running, and assume that is providing peek stress. Although we are measuring each service independently, the presences of service concurrency may have an overall impact on any service.

Ability to have lightweight test instances that can be positioned at strategic locations of the network such as in branch would be fundamental in having overall successful validation. Furthermore, there are recent efforts underway such as NetSecOPEN and ratification of RFC 9411 that can be helpful in establishing baseline open and transparent testing.

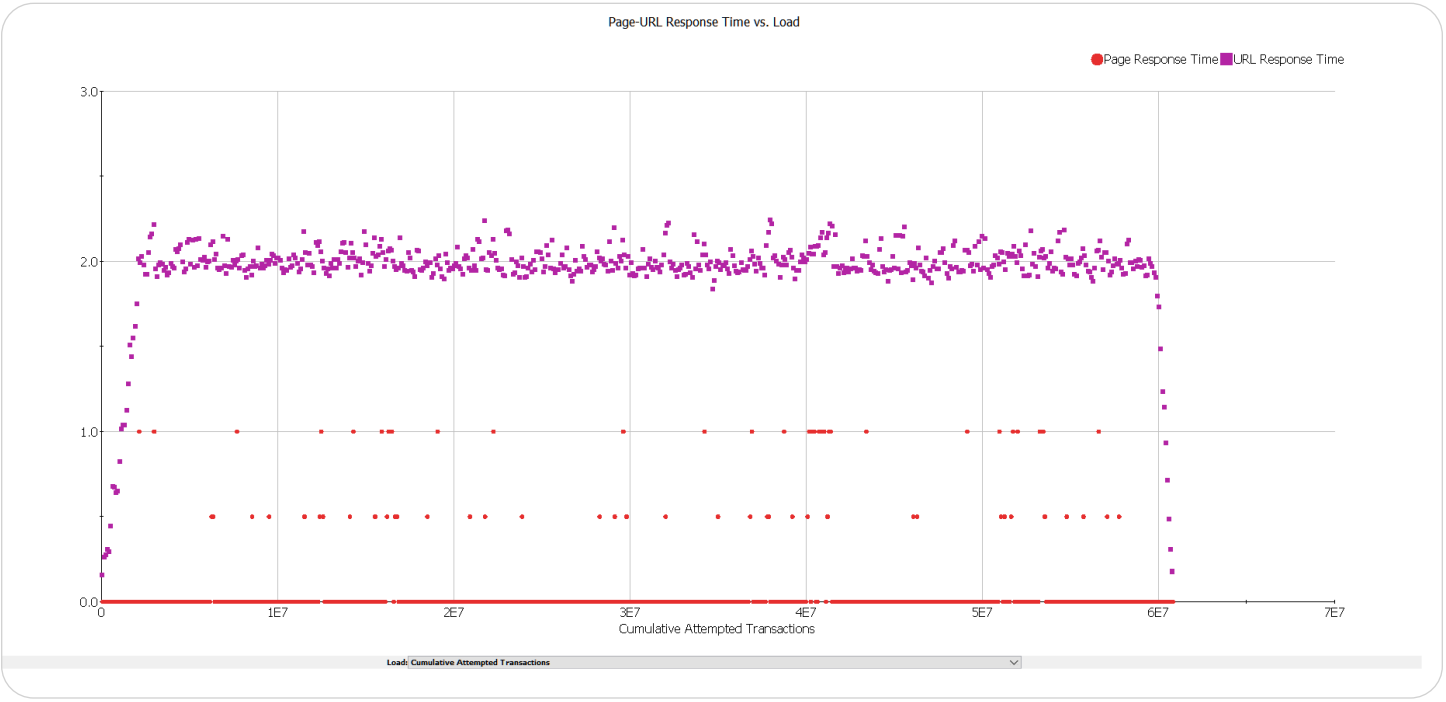


Correct KPI Measurement

How and what we measure in SASE and SD-WAN environments is as critical as the traffic pattern we use to load the SD-WAN circuit. Since we have established that bandwidth, CPS, and open connections are sub-metrics of overall measurement, but individually not sufficient to measure QoE, we must establish a hierarchy of what to measure.

Going back to the concept that critical services are the primary traffic model of generation for SASE/SD-WAN, the logical question of what to measure is obvious. How customers perceive quality of the services transport over circuits over time is the optimal unit of measure. When a user is interacting with a service, they are moving between pages in a sequence forming a scenario. Since most pages tend to have about the same number of URLs, bearing any special content inspection, a page is a page. We can use that meaningful page as a unit of measure.

For HTTPS-based services, the user experience measurement is dependent on three axes working together to assure acceptable quality. The first axis of quality is **total page load time** (or render time). This is the time it takes from pressing "Enter" on the URL bar to page fully rendering, so it includes all sub-objects on the page. For this KPI, we use milliseconds as the time-based metric for all bound page URLs to process. Since SSL/TLS protocol exchanges are also a cofactor, this total page load time is inclusive of SSL tunnel formation time, rekey, certificate processing, and SNI processing as well. As a rule of thumb, 2,000 milliseconds is considered excellent, 4,000 milliseconds is average and >7,000 milliseconds is poor.

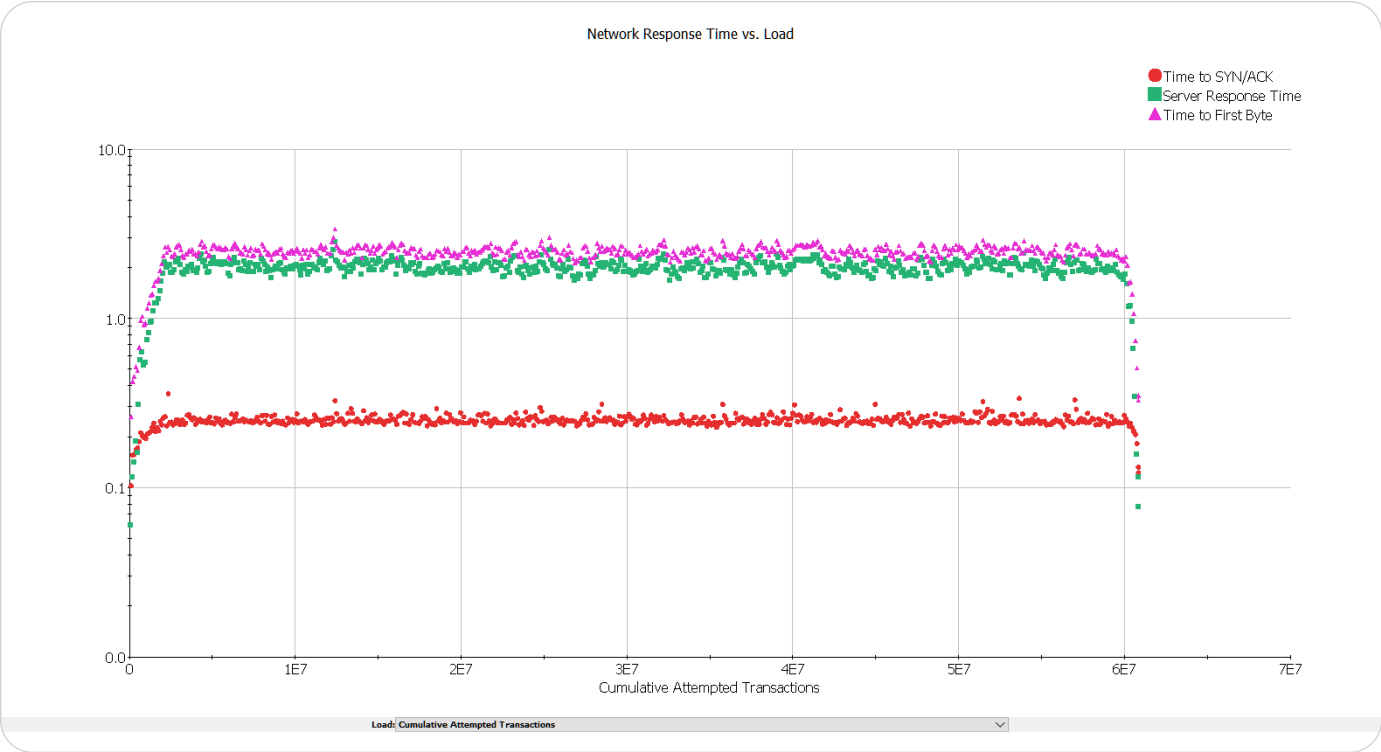


The next axis of quality is a binary metrics of **transaction error** (loss, timeout, corruption). The reason why TCP-based errors are not directly considered (such as timeout, retransmission, slow start, etc.), is because those are unobservable by the user directly, but contribute and effect the health of upper layer transactions. Depending upon the situation, the TCP error may be corrected without impact on experience. By using transactions, representing objects on the test page, any sufficiently extended TCP error will have an impact on transaction quality or latency which is concurrently being measured by the transaction metrics. The KPI for transaction error is count. The modern page must have zero transaction error.

The last KPI for quality is **page load variance** over a sufficiently large sample size. Over a very large sample (millions of full pages in the test), we examine the spread of page load time (in milliseconds as percent). Since this is a measure of predictability, one might experience a small variance initially, but because of degradation of the SD-WAN infrastructure, it may get measurably worse over time. It is critical that core services transported over SASE gateways and SD-WAN circuits have a predictable behavior. The KPI for variance is deviation as a percent from the median. Less than absolute 5% spread (+/- 2.5% on either side of the median) is desired. The advantage of looking at three axes of quality concurrently is that if all three align, no more measurement is necessary, and you will minimize false-positive results. Realistic traffic patterns paired with meaningful three-axis analysis gives you a powerful framework for independent testing.

For voice and video traffic, we use predefined MOS scores which are well established. For SIP-based voice, a PESQ MOS score of 4.2 or greater is preferred. Likewise, for classic video MOS, we want a score 4.0<=x<=5.0. Modern over-the-top video is transported over HTTPS. Since by definition there is no loss (because of TCP), we use an AS-score ranking scheme which is a normalized 0-100% scale of offered vs measured http goodput. Here, a score of 95%+ is desirable with only ABR upshifts and no downshifts. As with HTTPS, we suggest to also factor in variance with the same 5% spread previously discussed. The critical component is to have a statistically significant number of samples over time to increase the probability that test measurements will corelate to real-world scenarios. For voice and video, no fewer than a few hundred thousand streams should be used when performing long duration SOAK testing. The QoE metrics mentioned are "top of the pyramid" results.

Additional "under-the-hood" metrics should also be reported to enhance clarity of the SASE/SD-WAN device under test (DUT) metrics.



Test Load Pattern, Duration, and Minimum QoE Declaration

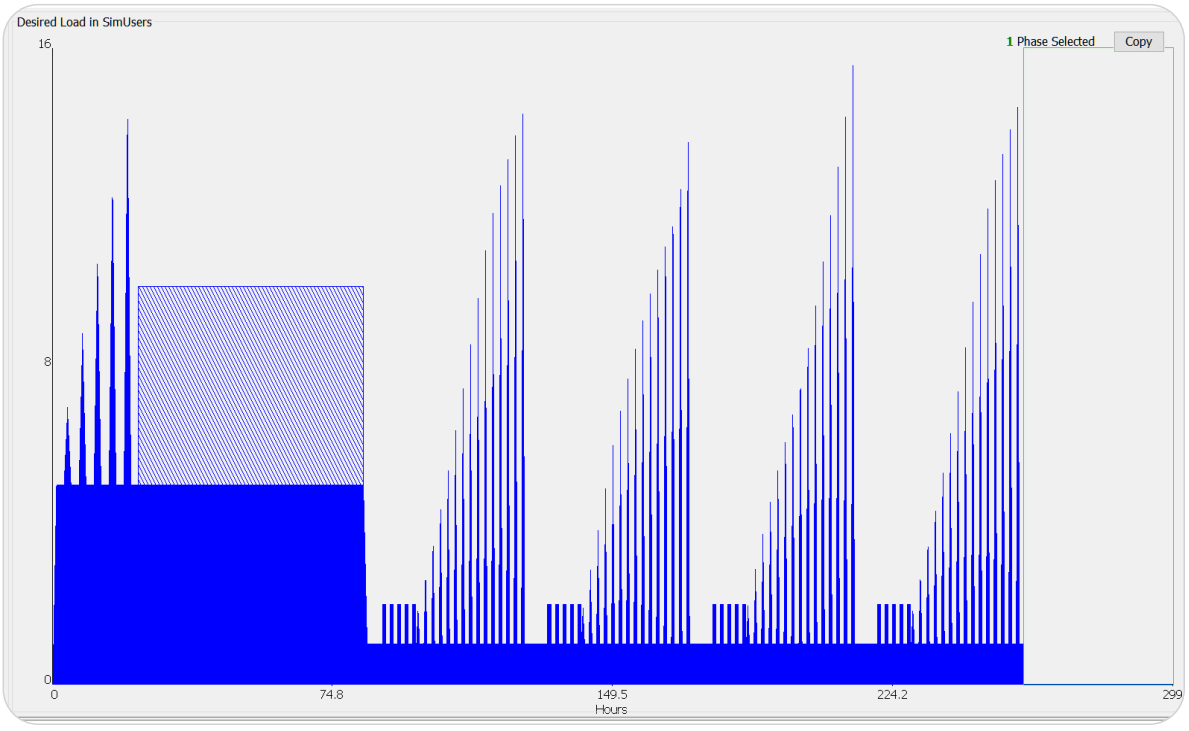
How traffic is loaded and for how long is also critical to rigorous SASE/SD-WAN testing. Loading patterns tend to fall into three main categories: baseline, customer modeling, and Soak. Depending on the objective of the test case, an appropriate loading pattern would be applied. Before we can apply a loading profile, we must define a minimum acceptable QoE measurement based on the services that we are testing. The form of this is a simple statement, "No customer will ever experience a QoE measurement worse than XYZ under any circumstance." In addition, your minimum QoE declaration may have differentiated services. If the SASE/SD-WAN allows for quality tiers or SLA levels, basically create one declaration statement for each level of how the policy of the environment is tiered and state how each level gives way to high service level congestion.

The **baseline pattern** tends to have a specific objective to characterize some part of the behavior of the network. For example, the tester may wish to put a complex list of action but only load a single use for a single pass to measure a baseline QoE measurement. These loading patterns tend to be very simple and can measure best case concurrency scale questions.

With the **customer modeling** loading profile, the objective is to model some extended period with the same degree of loading complexity as seen in a real customer environment. These loading profiles tend not to be "ramp up, sustain, ramp down" but factor in burst time period, randomness, periodicity, over a meaningful period of time of at least one business day. A typical example of this class of loading profile would be the 24-hour test, which might have hours of bursty traffic, random traffic and high load period over a 24-hour window.

The last class of loading profile, the **Soak** pattern, can be any pattern that repeats of over a very long period of time of continuous loading (> 24 hours, up to a week). This pattern is intended to measure any system in the SASE/SD-WAN DUT that may degrade over load.

The loading pattern and QoE declaration combine together to give you an accurate and meaningful measure of true scale. The QoE declaration will always limit the scale of load. In fact, best practice is when your QoE is being violated, you must scale back your concurrent or rate of load until QoE reconverges and stays converged. Likewise, if you have reached the top of your load profile and QoE is still converged, then you could potentially add more concurrency and achieve a better ROI on the infrastructure.



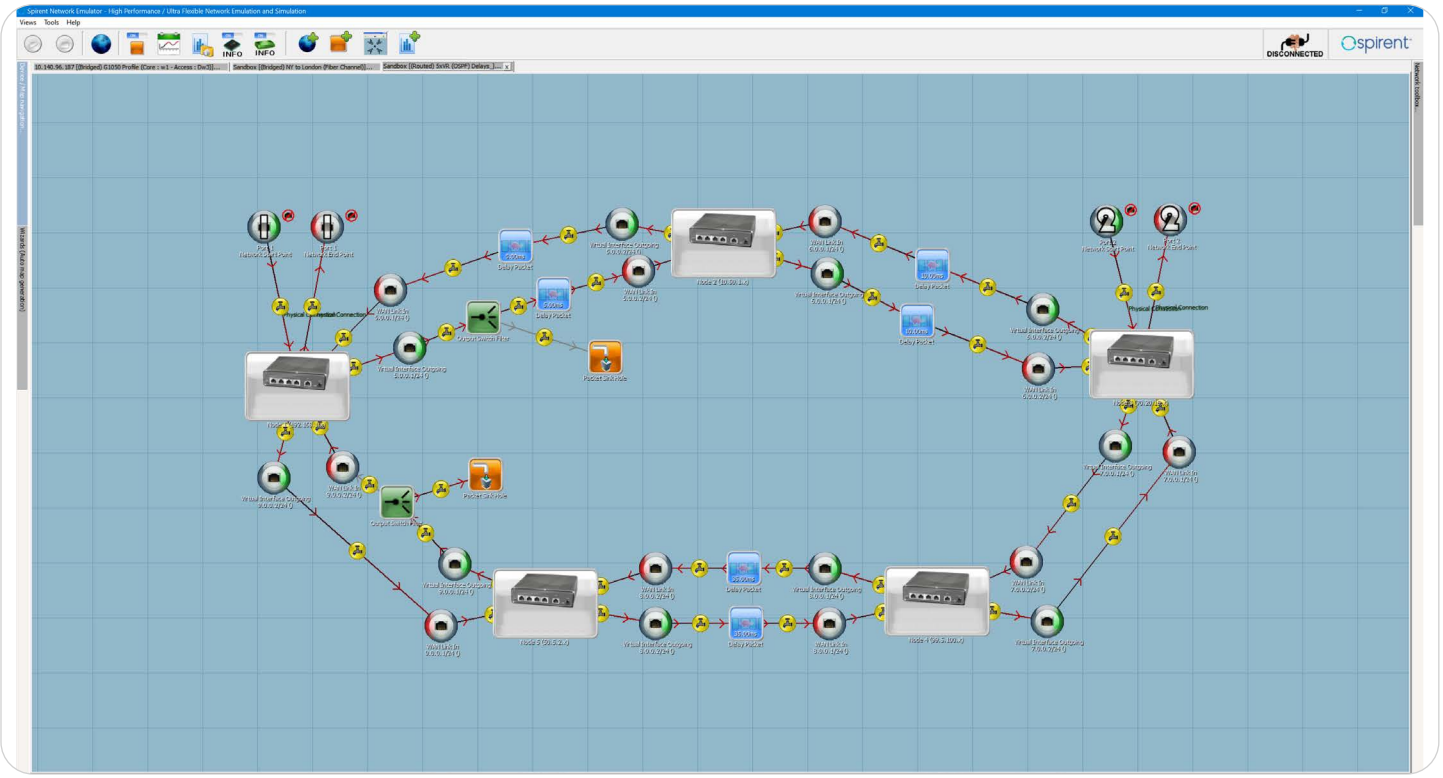
Effects of Physical WAN Elements on Quality of Experience

SD-WAN circuits are still WANs, and are affected by physics of distance as well as non-trivial WAN impairments. It is actually very important for correct methodology to include WAN effects in the test plan for SD-WAN circuits. The perception of quality users experience is all inclusive from the client to servers and any element in between in the chain. The means that there is a discrete, maximum "Impairment" budget end-to-end in the chain such that users still experience that desired level of quality based on their service level.

Testing without the consideration of effects such as distance (latency), jitter, and sequence errors will inadvertently give the device under test too much tolerance. For example, you may determine in the lab that a DUT can forward at 100 Gbps if you do not include WAN effect, but when deployed, you may only get 70-80 Gbps. Not including physical or WAN behavior will have the effects of giving you performance that is "oversubscribed."

A better technique is to place the test endpoint at different edge points on a real SD-WAN. This would include the effects of the WAN on traffic and is a good sanity check. However, this technique has its limitations. First, it does not scale to multiple endpoints and multiple service levels in the SD-WAN. Next, it is not reproducible. You are capturing a moment in time with each test case. Third, it does not lend itself well to combination and automation test cases.

The solution is to use a tool such as Spirent Network Emulator (SNE) for emulating real-world attributes of an SD-WAN in a multi-port, programmatic fashion. Using existing circuits, use a tool such as Spirent TestCenter Virtual (STCv) to measure target circuit latency, jitter, and sequence errors by circuit. Then, configure those circuits in the Network Emulator for testing. You can also add G.1050 WAN impairment over time models to cycle though hourly, daily or weekly changes in impairment. Finally, you can factor in different VLAN CoS or L3 DiffServ QoS service levels for more realistic modeling.



About Spirent

Spirent Communications (LSE: SPT) is a global leader with deep expertise and decades of experience in testing, assurance, analytics and security, serving developers, service providers, and enterprise networks. We help bring clarity to increasingly complex technological and business challenges. Spirent's customers have made a promise to their customers to deliver superior performance. Spirent assures that those promises are fulfilled.

For more information visit: www.spirent.com

COMPUTER CONTROLS

Computer Controls AG distributes electronic components, IoT applications, software, test and measurement solutions. We support our customers in selection, integration and maintenance, translating requirements into complete electronic solutions and customizing systems according to individual specifications. We are a qualified partner for solution-orientated cutting-edge technology.

Computer Controls AG
Industriestrasse 53
8112 Otelfingen
Switzerland
Phone: +41 (0) 44 308 66 66
E-mail: info@ccontrols.ch
Web: https://www.ccontrols.ch

Summary

The selection of the right traffic pattern unit for testing, the emulation of the WAN attributes and security policies, vulnerability testing, and correct KPI measurements for SASE and SD-WAN traffic help minimize test durations and help the tester avoid incorrect assessment of real-world behavior.

QoE is the best unit of measure giving real world traffic stimulus, because it is a direct measure of user satisfaction. It is based on performance, detection of errors, and variability for SSL/TLS based services and MOS scores for video and voice.

This pattern is also optimal for changing underlay that customer flows will experience across the SASE and SD-WAN. Because the traffic is not only real but of sufficient complexity to pass even content aware routing and DPI services, there will be no need for the tester to lower the SD-WAN circuit feature set. Lastly, there is high confidence that every element is fully tested in the SASE and SD-WAN service chain.

Americas 1-800-SPIRENT
+1-800-774-7368 | sales@spirent.com

Europe and the Middle East
+44 (0) 1293 767979 | emeainfo@spirent.com

Asia and the Pacific
+86-10-8518-2539 | salesasia@spirent.com

Rev E | 02/24 | www.spirent.com