

# RFC 6349 Testing with TrueSpeed from VIAVI Solutions — Experience Your Network as Your End-Users Do

RFC 6349 is a transmission control protocol (TCP) throughput test methodology that VIAVI co-authored along with representatives from Bell Canada and Deutsche Telecom. Issued by the Internet Engineering Task Force (IETF) organization, RFC 6349 provides a repeatable test method for TCP throughput analysis with systematic processes, metrics, and guidelines to optimize the network and server performance.

This application note summarizes RFC 6349, “Framework for TCP Throughput Testing,” and highlights the automated and fully compliant VIAVI RFC 6349 implementation, TrueSpeed™, now available on the T-BERD/MTS-5800 Handheld Network Tester, T-BERD/MTS 5800-100G, MAP-2100, and VIAVI Fusion, a virtual network test and assurance system.

This application note also discusses the integration of TrueSpeed RFC 6349 with the ITU Y.1564 Ethernet service activation standard. This powerful testing combination provides a comprehensive means to ensure an optimized end-customer experience in multi-service (such as triple play) environments.

## RFC 6349 TCP Test Methodology

RFC 6349 specifies a practical methodology for measuring end-to-end TCP throughput in a managed IP network with a goal of providing a better indication of the user experience. In the RFC 6349 framework, TCP and IP parameters are also specified to optimize TCP throughput.

RFC 6349 recommends always conducting a Layer 2/3 turn-up test before TCP testing. After verifying the network at Layer 2/3, RFC 6349 specifies conducting the following three test steps.

- Path MTU detection (per RFC 4821) to verify the network maximum transmission unit (MTU) with active TCP segment size testing to ensure that the TCP payload remains unfragmented
- Baseline round-trip delay and bandwidth to predict the optimal TCP window size for automatically calculating the TCP BDP
- Single and multiple TCP connection throughput tests to verify TCP window size predictions that enable automated “full pipe” TCP testing

The following subsections provide details for each RFC 6349 test step.

## Path MTU Discovery (Per RFC 4821)

TCP implementations should use path MTU discovery techniques (PMTUD) which rely on Internet control message protocol (ICMP) “need to frag” messages to learn the path MTU. When a device has a packet to send that has a don’t fragment (DF) bit in the IP header set and the packet is larger than the MTU of the next hop, the packet is dropped and the device sends an ICMP need to frag message back to the host that originated the packet. The ICMP need to frag message includes the next-hop MTU, which PMTUD uses to adjust itself. Unfortunately, because many network managers completely disable ICMP, this technique can be somewhat unreliable.

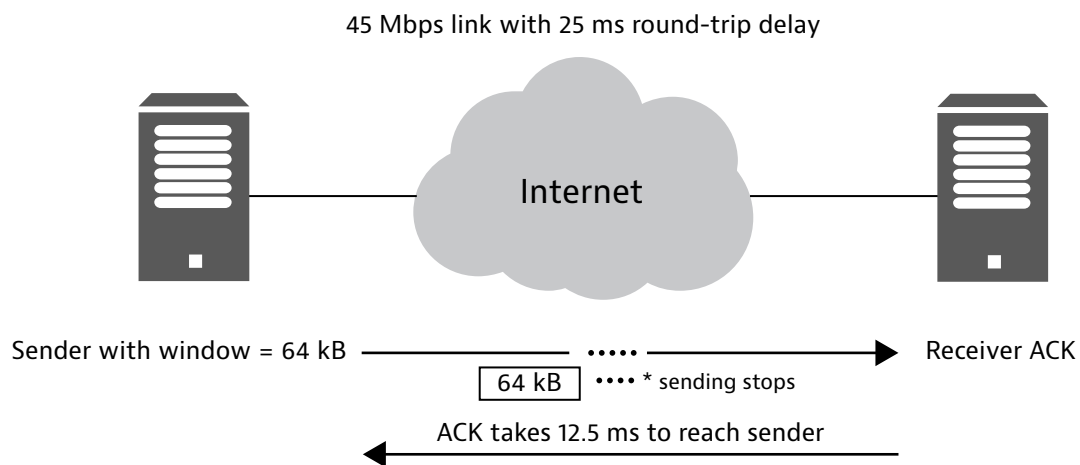
Therefore, RFC 6349 suggests conducting packetization-layer path MTU discovery (PLPMTUD) per RFC 4821 to verify the network path MTU because it can be used with or without ICMP. PLPMTUD specifies that live TCP traffic is used to poll the network for the MTU. The same technique of setting the DF bit of the IP packet is implemented, but it does not rely on ICMP, because it uses a live TCP session. The algorithm uses TCP retransmit conditions to search for the MTU, which is used to avoid fragmentation in all subsequent steps.

## Baseline Round-Trip Delay and Bandwidth

Before TCP testing can begin, it is important to determine the baseline round-trip time (RTT), or the noncongested inherent delay, and bottleneck bandwidth (BB) of the end-to-end network. These baseline measurements are used to calculate the BDP and to provide estimates for the sizes of TCP receive window (RWND) and send socket buffer that will be used in subsequent test steps.

On a wide-area network (WAN) link, TCP must be properly configured to adjust the number of bytes the sender can transmit before receiving an acknowledgment (ACK) from the receiver. This number of bytes “in-flight” is commonly referred to as the TCP window; although, in reality, there are several TCP window mechanisms at work.

Figure 1 depicts the concept of the TCP in-flight data bytes on a 45 Mbps WAN link with 25 ms round-trip delay (RTD), or latency.



1107.0422

Figure 1. Illustration of TCP in-flight data bytes on a 45 Mbps WAN link with 25 ms RTD

In Figure 1, the TCP window is improperly tuned and only 64 kB are transmitted from the sender before requiring an ACK.

As RFC 6349 describes, the BDP is the optimum TCP window, calculated as:

$$\text{BDP} = \frac{\text{link bottleneck bandwidth} \times \text{round-trip time}}{8}$$

In this example, the BDP would be 140 kB, which is more than twice the size of the sender's 64 kB window and the sender would only achieve about 20 Mbps throughput.

RFC 6349 defines these mechanisms for measuring the RTT:

- Active traffic generation at Layer 2/3 and a loopback from one end to the other
- Packet captures
- Extended management information bases (MIBs) (RFC 4898) from network devices
- ICMP pings

The BDP depends on both the RTT and the BB, so it requires also measuring BB. Layer 2/3 testing, such as RFC 2544, adopted for operational networks, is specified as one means for measuring the BB. Once both the RTT and BB are known, RFC 6349 enables computation of the expected TCP performance for subsequent TCP throughput tests.

### Single and Multiple TCP Connection Throughput Tests

Deciding whether to conduct single- or multiple-TCP connection tests depends upon the size of the BDP in relation to the TCP RWND configured in the end-user environment. For example, if the BDP for a long fat network (LFN) is 2 MB, then it is probably more realistic to test this network path with multiple connections. Assuming typical host TCP RWND sizes of 64 kB (for example, Windows XP) using 32 TCP connections would emulate a small-office scenario.

While RFC 6349 does not mandate testing multiple connections, it is strongly recommended as the most realistic means for accurately verifying TCP throughput. RFC 6349 also defines specific metrics to measure during TCP throughput tests, which is discussed next.

## RFC 6349 Metrics

The following presents RFC 6349 TCP metrics along with examples for using them to diagnose causes for suboptimal TCP performance.

### TCP Transfer Time

The first RFC 6349 TCP metric is the TCP transfer time, which simply measures the time it takes to transfer a block of data across simultaneous TCP connections. The ideal TCP transfer time is derived from the network path BB and the various Layer 1/2/3 overheads associated with the network path, for example, the bulk transfer of 100 MB upon five simultaneous TCP connections over a 500 Mbps Ethernet service, each connection uploading 100 MB. Each connection may achieve different throughput during a test, therefore, determining the overall throughput rate is not always easy, especially as the number of connections increases.

The ideal TCP transfer time is approximately 8 seconds, however, in this example, the actual TCP transfer time was 12 seconds. The TCP transfer index would be  $12 \div 8 = 1.5$ , indicating that the transfer across all connections took 1.5 times longer than the ideal.

## TCP Efficiency

TCP retransmissions are normal phenomena in any TCP/IP network communication. Determining the number of retransmissions that will impact performance is difficult when simply using the number itself. RFC 6349 defines a new metric to gain insight into the relative percentage of a network transfer that was used due to the retransmission of a payload.

This metric is the TCP Efficiency metric, or the percentage of bytes not retransmitted, and is defined as:

$$\frac{\text{transmitted bytes} - \text{retransmitted bytes}}{\text{transmitted bytes}} \times 100$$

Transmitted bytes are the total number of TCP payload bytes transmitted including the original and retransmitted bytes. This metric provides a comparison between various quality of service (QoS) mechanisms such as traffic management, congestion avoidance, and various TCP implementations, such as Reno and Vegas to name a few.

For example, if 100,000 bytes were sent and 2,000 had to be retransmitted, the TCP Efficiency would be calculated as:

$$\frac{102,000 - 2,000}{102,000} \times 100 = 98.03\%$$

Note that packet loss percentages at Layer 2/3 do not directly correlate to retransmission percentages of bytes because the distribution of the packet loss can widely affect the manner in which TCP retransmits.

## Buffer Delay Percentage

RFC 6349 also defines the Buffer Delay Percentage, which represents the increase in RTT during a TCP Throughput test from the baseline RTT, which is the RTT inherent to the network path without congestion.

The Buffer Delay Percentage is defined as:

$$\frac{\text{average RTT during transfer} - \text{baseline RTT}}{\text{baseline RTT}} \times 100$$

For example, use the following formula to calculate the Buffer Delay Percentage of a network with a baseline RTT path of 25 ms that increases to 32 ms during an average RTT TCP transfer.

$$\frac{32 - 25}{25} \times 100 = 28\%$$

In other words, the TCP transfer experienced 28-percent additional RTD (congestion) which may have caused a proportional decrease in overall TCP throughput leading to longer delays for the end user.

## RFC 6349 TCP Tuning Guidelines

For cases where the TCP performance does not meet expectations, RFC 6349 provides guidelines for possible causes.

- Intermediate network devices can actively regenerate the TCP connection and can alter TCP RWND size, MTU, and other things
- Rate limiting by policing instead of shaping causes excessive TCP retransmission due to tail drops

- Maximum TCP buffer space  
All operating systems have a global mechanism that limits the amount of system memory used by TCP connections. On some systems, each connection is subject to a memory limit that is applied to the total memory used for input data, output data, and controls. On other systems, separate limits exist for input and output buffer spaces per connection. Client/server IP hosts might be configured with maximum TCP buffer space limits that are far too small for high-performance networks.
- Socket buffer sizes  
Most operating systems support separate per-connection send-and-receive buffer limits that can be adjusted within the maximum memory limits. These socket buffers must be large enough to hold a full BDP of TCP bytes plus overhead. Several methods can be used to adjust the socket buffer size, but TCP auto-tuning automatically adjusts these as needed for optimal balance of TCP performance and memory usage.

Refer to RFC 6349 for the complete list of network/host issues and recommended solutions.

### VIAVI Implementation of RFC 6349

VIAVI has integrated the RFC 6349 test method into its multiple portable network test devices, a rack-mounted test device, and a software-based network test system. TrueSpeed uses test configuration files so that technicians can simply load a test configuration, press “go,” and publish a test report with results.

Figure 2 illustrates a scenario using the VIAVI TrueSpeed test capability.

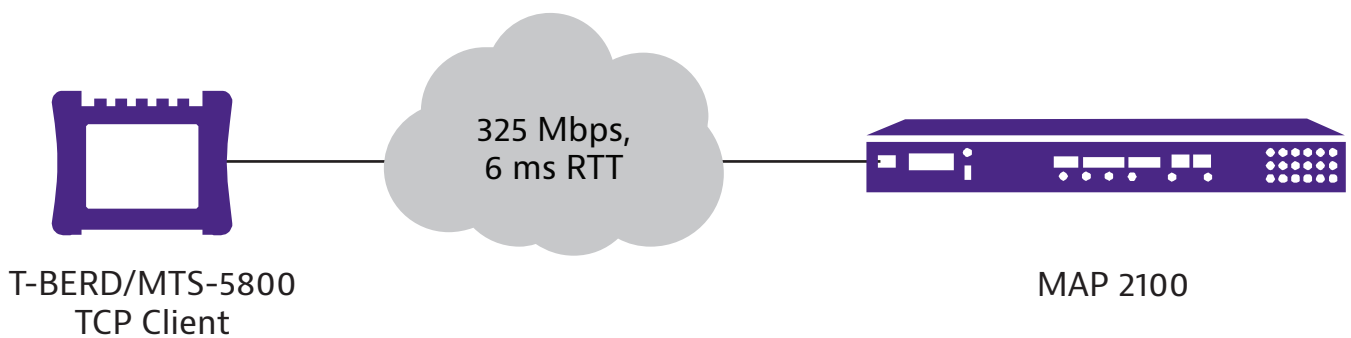


Figure 2. Test scenario for TrueSpeed throughput testing

1106.0422

This is an LFN with a customer-committed information rate (CIR) of 325 Mbps, an RTT of ~6 ms, and a BDP of ~250 kB. In this example, the T-BERD/MTS-5800 acts as a TCP client that conducts upload throughput tests to the TCP server, which is a T-BERD/MTS-6000A.

The test then runs automatically and completes in an average of 3 minutes using the recommended default settings. Each test step provides graphical results.

Tests run in the order specified in RFC 6349 with the first being the Path MTU test. Figure 11 shows the test result for this test using our example network with a Path MTU of 1500 bytes.

There are two workflows for the TrueSpeed test:

- Installation Test Mode: the user is required only to enter addressing and CIR value. The T-BERD/MTS automatically populates all TCP parameters per RFC 6349
- Troubleshooting Test Mode: the more advanced user can control many aspects of the TCP test to perform focused analysis that also includes an advanced traffic-shaping test

The following topics summarize the two different test modes.

### Installation Test Mode

In this mode, the technician is dispatched to provision/install a new end-customer service and would run RFC 2544 or Y.1564 Layer2/3 test first. Then, using all of the same T-BERD/MTS addressing information (for example, IP addresses, VLAN, QoS) to conduct the automated TrueSpeed installation test.

With a remote T-BERD/MTS configured with an IP address, all testing is conducted from a local T-BERD/MTS (one-person RFC 6349 test). The following is an overview of the test sequence.

The technician enters CIR and test time.

- T-BERD/MTS automatically populates all fields for TCP window size and connection count
- T-BERD/MTS runs upload then downloads (speed test) from the local unit
- Reports a simple pass/fail and report to local T-BERD/MTS.

A more detailed step-by-step guide is represented below along with T-BERD/MTS reference screenshots.

1. The technician configures the IP address (and VLANs if used) for the local and remote T-BERD/MTS and then a ping can also be issued to verify Layer 3 connectivity.

The local T-BERD/MTS connects to the remote T-BERD/MTS and uses TCP port 3000 for all test configuration and results retrieval.

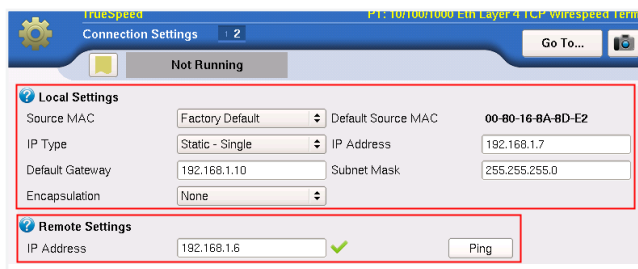


Figure 3. IP address configuration

2. The technician configures one screen to test the SLA at Layer 4 as shown below:

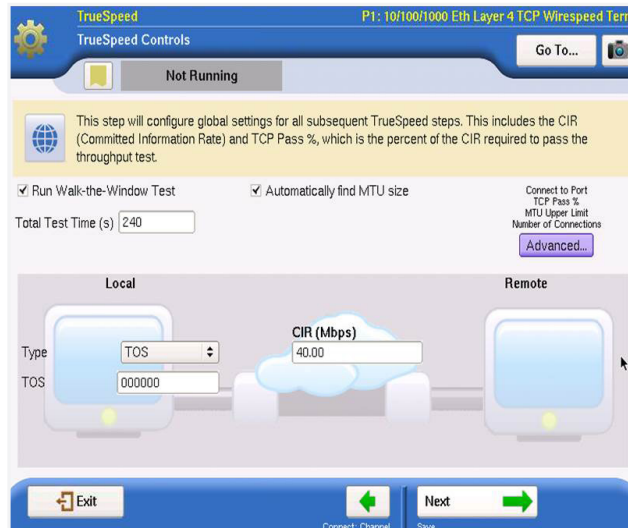


Figure 4. SLA test configuration

1. Total test time for all TCP tests (minimum is 30 seconds).
2. Local and remote QoS/VLAN settings (VLAN not shown).
3. Layer ½ CIR for the service to be tested.

There are no complex TCP Window sizes to configure or number of connections. The T-BERD/MTS uses RFC 6349 to auto-compute these values for the user.

3. The technician clicks **Run Test**.

The local T-BERD/MTS automatically conducts the RFC 6349 test in both the upstream and downstream direction (sequentially, like a speed test).

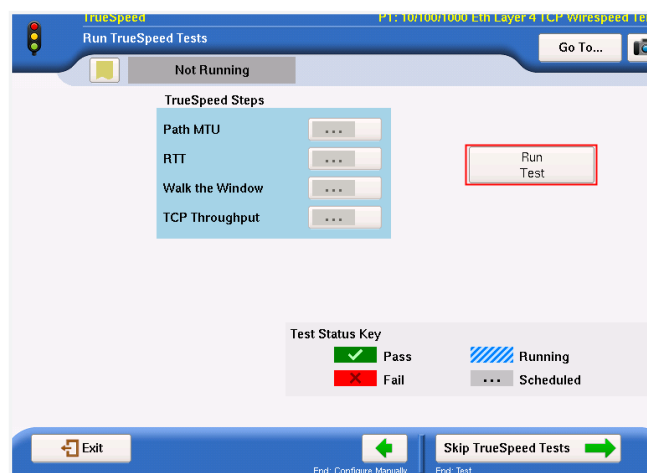


Figure 5. Running RFC 6349 test

The following tests are run per RFC 6349 with a brief description below; a more detailed description is provided in the following Troubleshooting Test Mode topic.

- Path MTU Detection (per RFC 4821) — verifies network MTU with active TCP segment size testing to ensure TCP payload does not get fragmented
- RTT test — measures RTT of the service and predicts optimum TCP window size to automatically calculate the TCP BDP
- Walk-the-Window — conducts four different TCP Window size tests and ramps the throughput from 25% to 100% of Layer 4 CIR
- TCP Throughput — conducts a more detailed throughput test at the CIR and provides a pass/fail verdict, RFC 6349 metrics, and detailed graphs

The results of the Walk-the-Window tests are shown and are accessed by clicking on the box next to the result.

Notice that there is an Upstream and Downstream button for the tests. In this example, the Upstream had a 40 Mbps policer and had dramatic performance issues with all window settings. The CIR window setting is always the fourth window tested, which in this case, should have produced a result of 40 Mbps.

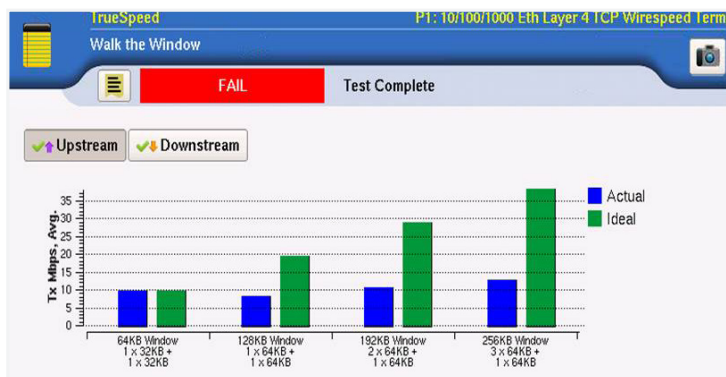


Figure 6. Walk-the-Window test screen — Upstream

In Figure 7, there was no policer in the Downstream direction and the throughput met the ideal in every case, including the fourth window size (which equaled the CIR window size).

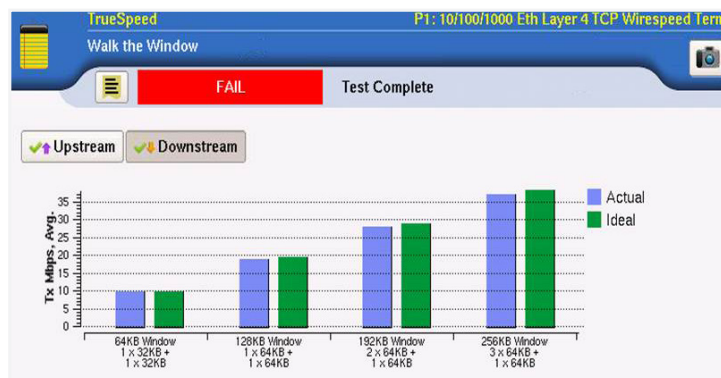


Figure 7. Walk-the-Window test screen — Downstream



As previously mentioned, the TCP Throughput test is conducted at the CIR Window size (4th of the Walk the Window series) and provides a more detailed, longer test.

After test completion, the user is presented a simple pass/fail verdict (Figure 8) along with a detailed throughput test result screen (Figure 9); in this example, the test failed in the Upstream direction due to the 40 Mbps policer. The actual customer throughput would be only 12.3 Mbps under this condition. In addition, the TCP Efficiency and Buffer Delay metrics help diagnose the cause of the poor TCP performance. In this example, the policer is dropping packets.

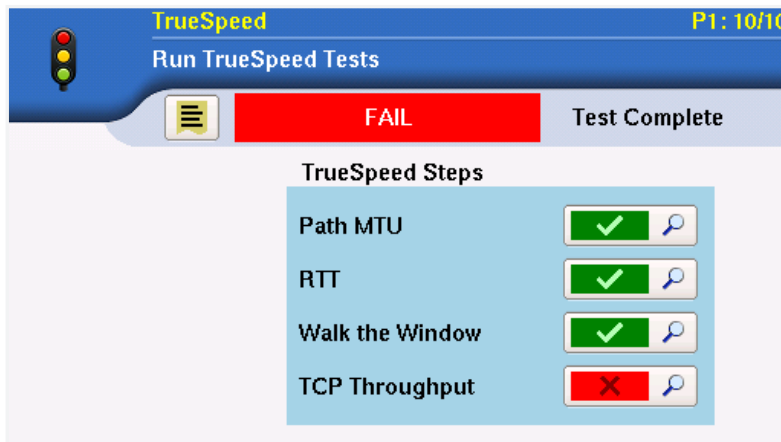


Figure 8. Pass/fail test results

After the test has completed, a graphical test report is produced and the test configuration can also be saved.

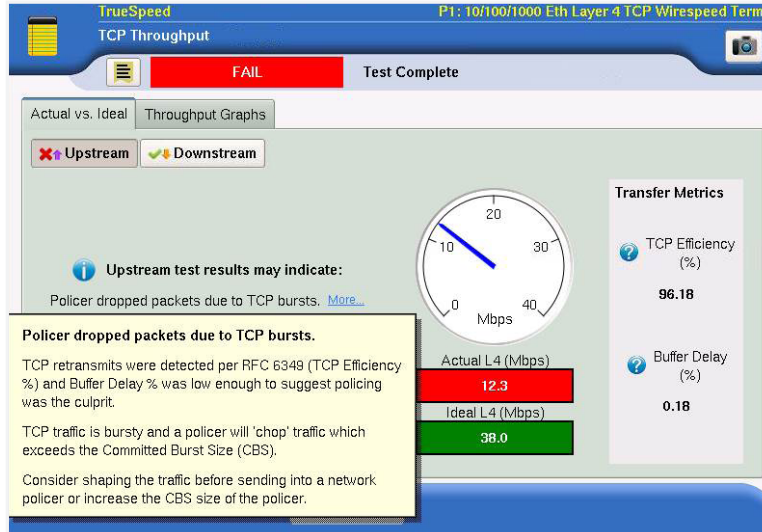


Figure 9. Detailed TCP-throughput test results

## Troubleshooting Test Mode

In this mode, a user can also either load a test configuration or manually configure the test. This mode is highly configurable for the advanced field technician and a more detailed test scenario is explored with a more-detailed explanation of TCP theory and RFC 6349 results.

The user can execute all RFC 6349 test steps or a subset of these tests as Figure 10 illustrates. In this example, the CIR is 325 Mbps and RTT is 6.5 ms.

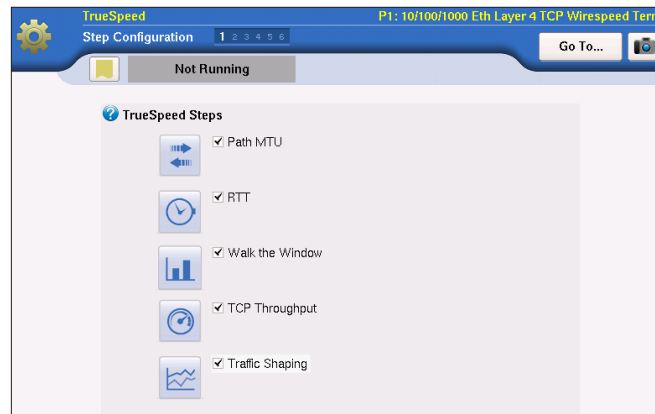


Figure 10. TrueSpeed test configuration setup

The test then runs automatically and completes in an average of 3 minutes using the recommended default settings. Each test step provides graphical results.

Tests run in the order specified in RFC 6349 with the first being the Path MTU test. Figure 11 shows the test result for this test using our example network with a Path MTU of 1500 bytes.

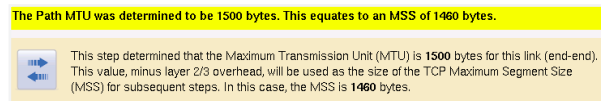


Figure 11. Path MTU test results

After completing the Path MTU test, TrueSpeed proceeds to the RTT test which is essential because BDP dictates the ideal TCP window. The BDP is used in subsequent test steps to predict ideal TCP throughput.

Figure 12 shows the RTT test result for this example with an RTT of 6.5 ms.

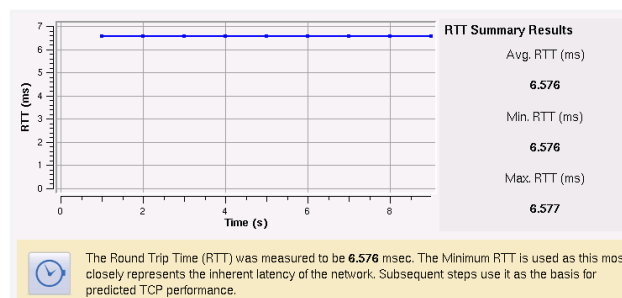


Figure 12. RTT test results

The Walk the Window test provides an informative characterization of tested window size results and expected results. The Walk the Window test uses the parameters from the path MTU and RTT tests to conduct the window size throughput tests. Figure 13 shows results for the Walk the Window test.

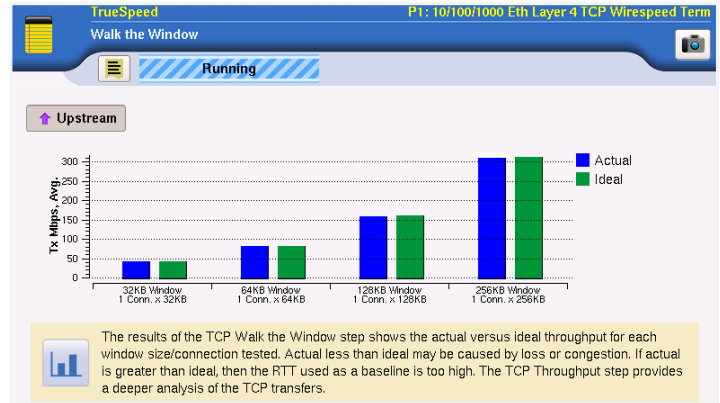


Figure 13. Walk the Window test results

In the example in Figure 13, the actual TCP throughput would only saturate the CIR of 325 Mbps with a TCP window size configured to 256 kB. Many times, end-host computers use much smaller windows, such as 64 kB, resulting in much lower than expected throughput. Here, a 64 kB window only achieved ~80 Mbps.

Next, the TCP Throughput test allows for detailed analysis of a problematic window size and provides the RFC 6349 metric results to assist in the diagnosis. In Figure 10, the TCP window was increased to 384 kB (using three connections of size 128 kB), which significantly oversubscribes the 325 Mbps CIR. End users often go to this extreme thinking, “the larger the window the better”. However as this WAN environment shows in Figure 14, network policing activated at the 325 Mbps CIR and significantly degraded TCP performance.

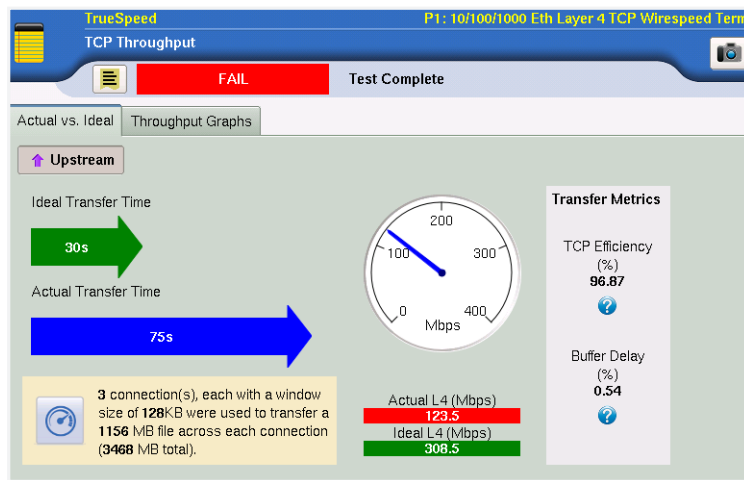


Figure 14. TCP Throughput test results (basic view)

Here, the TCP Efficiency metric of 96.87 percent and the Buffer Delay Percentage of only 0.54 percent indicates that loss rather than a buffering delay caused the performance gap. Figure 15 shows more detailed examination of the throughput graphs.

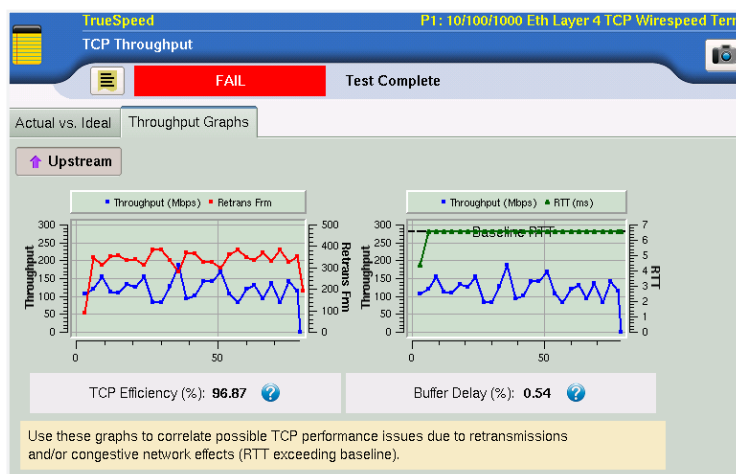


Figure 15. TCP-throughput test graphs

VIAVI extends RFC 6349 testing and provides a traffic shaping test. Traffic shaping is intelligent network buffering, where the network device shapes the traffic according to the CIR. Traffic shaping should be performed at the customer premises equipment (CPE) edge device, but network providers also can shape traffic to substantially benefit TCP performance and the end-customer experience.

By not shaping TCP traffic as it downshifts from a higher speed interface to a lower speed, network policers can detrimentally affect TCP performance. Contrary to shaping, policing chops excess traffic above the CIR, causing TCP retransmissions and seriously degrades end-user performance. Figure 16 contrasts the function of a traffic shaper versus a policer.

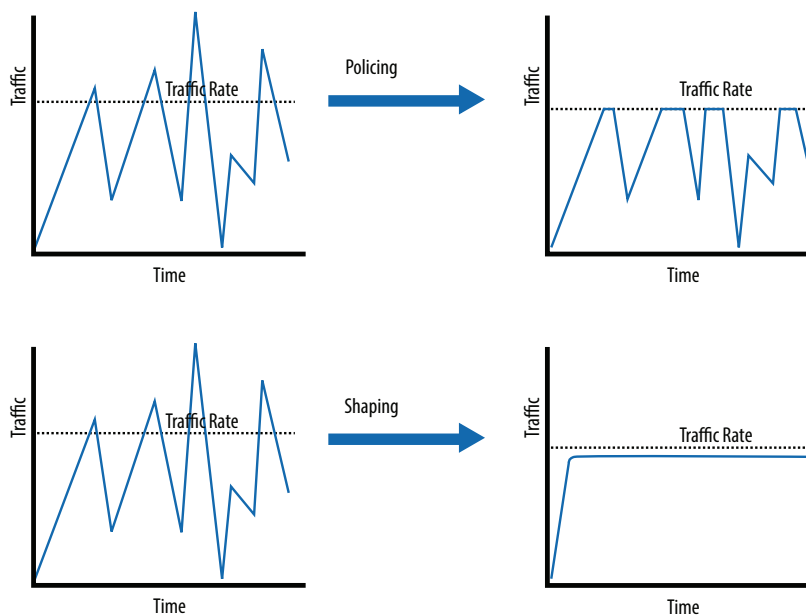


Figure 16 Function of a traffic shaper versus a policer

TrueSpeed provides a traffic-shaping test result that clearly shows traffic that is being shaped versus policed. Figure 17 shows traffic that is being policed and has a very jagged distribution of bandwidth among four TCP connections.

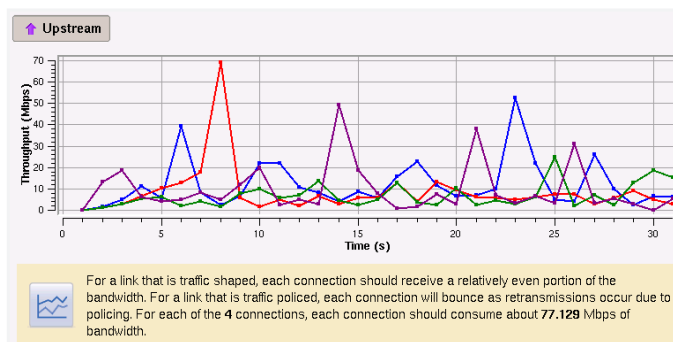


Figure 17. TrueSpeed Traffic Shaping result (where traffic is policed)

Figure 18 shows traffic shaping with very even distribution of bandwidth among four TCP connections.

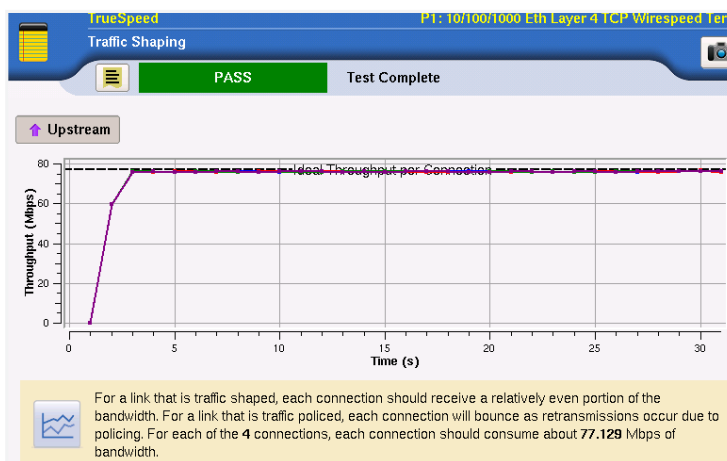


Figure 18. TrueSpeed Traffic Shaping result (where traffic is shaped)

# Integrating TrueSpeed RFC 6349 with Y.1564

ITU Y.1564 is an ITU standard for Ethernet service activation. Highlights include:

- Multiple services field turn-up and installation test to meet customer SLAs
- Automated end-to-end, multi-Ethernet/IP service test using loopback on the far end
- Ideal for LTE/4G IP services and triple-play testing.

Problems detected by Y.1564 include:

- Network misconfigurations — VLAN ID and priority, IP TOS, max throughput
- Poor quality of service — too much latency, jitter, or loss
- Services not working well together on the same network under load conditions.

Since Y.1564 is only defined to verify Layer 2 (Ethernet) and Layer 3 (IP) performance, the testing gap at the TCP layer is untested. The net result is that a Y.1564 can provide “passing” results and yet the performance for the end customer can still be poor due to TCP-related performance issues defined in previous sections.

The solution to this testing deficiency is to integrate TrueSpeed RFC 6349 testing with Y.1564 during service activation. Figure 19 illustrates how TrueSpeed can be integrated with the Y.1564 service performance test.

In Figure 19, voice and video services are tested as constant bit rate, UDP-based streams. However, the data service is tested with TrueSpeed RFC 6349 compliant traffic which is TCP based and bursty. The bursty nature of TCP applications can stress network QoS and cause performance issues that remain undetected when running a pure Y.1564 test.



Figure 19. Y.1564 Performance test phase with integrated TrueSpeed service

The VIAVI implementation of this integrated approach is called SAMComplete and it is the industry’s only service-activation methodology to integrate RFC 6349 with Y.1564. SAMComplete provides an automated configuration of the TrueSpeed service. Users need only specify the CIR and SAMComplete will automatically configure the appropriate number of TCP sessions for the network conditions. At the end of this integrated test, users are provided with a simple pass/fail status for the TrueSpeed service, just like the traditional Y.1564 services, as shown in Figure 20.

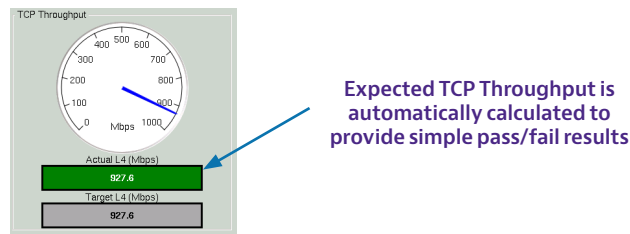


Figure 20. Simple pass/fail verdict for TrueSpeed RFC 6349 test

## Conclusion

This application note summarizes the TCP test methods specified in RFC 6349 that can remove significant variability in TCP testing methods with a step-by-step, best-practices approach for TCP throughput testing. The TCP metrics specified within RFC 6349 provide objective measures of network issues (loss and delay) and how they affect overall TCP performance.

In cases where actual TCP throughput does not equal the ideal, RFC 6349 provides practical guidelines for tuning the network and/or end hosts.

The VIAVI TrueSpeed test is a completely automated RFC 6349-compliant implementation that even novice technicians can perform in as few as five minutes because of its simple, push-button execution and automated reporting capability that more experienced network engineers can use to verify and implement SLAs.



Contact Us **+1 844 GO VIAVI**  
(+1 844 468 4284)

To reach the VIAVI office nearest you,  
visit [viasolutions.com/contact](https://viasolutions.com/contact)

© 2022 VIAVI Solutions Inc.  
Product specifications and descriptions in this  
document are subject to change without notice.  
Patented as described at  
[viasolutions.com/patents](https://viasolutions.com/patents/rfc6349-an-tfs-ae)  
rfc6349-an-tfs-ae  
30173007 904 0422

[viasolutions.com](https://viasolutions.com)